The Flower Calculus

Pablo Donato
LIX, École polytechnique, France

Abstract

We introduce the flower calculus, a deep inference proof system for intuitionistic first-order logic inspired by Peirce's existential graphs. It works as a rewriting system over inductive objects called "flowers", that enjoy both a graphical interpretation as topological diagrams, and a textual presentation as nested sequents akin to coherent formulas. Importantly, the calculus dispenses completely with the traditional notion of symbolic connective, operating solely on nested flowers containing atomic predicates. We prove both the soundness of the full calculus and the completeness of an analytic fragment with respect to Kripke semantics. This provides to our knowledge the first analyticity result for a proof system based on existential graphs, adapting semantic cut-elimination techniques to a deep inference setting. Furthermore, the kernel of rules targetted by completeness is fully invertible, a desirable property for both automated and interactive proof search.

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Constructive mathematics

Keywords and phrases deep inference, graphical calculi, existential graphs, intuitionistic logic, Kripke semantics, cut-elimination

Supplementary Material The source code for a Coq mechanization of additional meta-theoretical results, as well as a web demo of GUI for ITPs based on this work, are available as follows:
Software (Mechanized Theory): `https://github.com/Champitoad/flowers-metatheory` [16]
Software (Online Demo): `https://github.com/Champitoad/flower-prover` [17]

Acknowledgements I want to thank Luc Chabassier for writing the Lua script that was used to generate all the flower drawings in this document, and Benjamin Werner for useful feedback on a first draft of this paper. Lastly, I thank Tito for teaching me some invaluable formatting tricks.

1 Introduction

Graphical proof building Proof assistants — also called *interactive theorem provers* (ITPs) — provide a set of tools to ease the process of formalizing mathematical developments. This includes languages to specify definitions and statements conveniently, but also interfaces to build proofs interactively without having to fill in all the details. The dominant paradigm for these interfaces is that of *tactic languages* [40]: the user is exposed with a set of *goals* that remain to be proved, constituting the *proof state*, and modifies these goals through textual commands, called *tactics*, until there is no goal left. This is currently what is implemented in mainstream proof assistants such as Coq [54] and Lean [41].

In recent years, there have been several efforts to replace or complement textual tactic languages with *graphical user interfaces* (GUIs) [47, 4, 34, 12, 49, 31, 60, 3]. The hope is to make proof assistants more intuitive and accessible to beginners and non-specialists, but also, to some extent, more productive and ergonomic even for experts.

The initial motivation for this work was to design a proof calculus well-suited to *direct manipulation* in such a graphical setting. The idea is that the user should be able to interact directly with the graphical representation of the proof state, using a pointing device such as a mouse or fingers on a touch screen. In previous work [18], we proposed a way to synthesize complex logical inferences through *drag-and-drop* actions between two items of the current goal, based on the *subformula linking* (SFL) methodology [12, 13]. Since goals are represented as *sequents* $\Gamma \Rightarrow C$ in most ITPs, the items involved were traditional logical formulas, either two hypotheses in Γ or one hypothesis and the conclusion C.

Diagrammatic reasoning In this work, we show that (single-conclusion) sequents and symbolic formulas built from binary connectives and unary quantifiers are not mandatory for representing the proof state. Other authors have defended the idea of using *diagrams* as a more user-friendly frontend for ITPs. In particular, Linker et al. showed how to integrate tactic-based automation in an ITP based on *spider diagrams* [31], which are equivalent in expressive power to classical monadic *first-order logic* (FOL) [25].

We introduce a new data structure for goals inspired by an earlier invention in the history of diagrammatic logic: the *existential graphs* (EGs) of C. S. Peirce [48]. We noticed that our structure could be drawn and manipulated metaphorically in the form of nested *flowers*, and thus chose to name *flower calculus* the proof system for full intuitionistic FOL that we built around it. Our focus in this paper will be to introduce the flower calculus to readers unfamiliar with EGs, and to study its fundamental properties through the lens of modern *structural proof theory*.

Implementation We have formalized in Coq a bidirectional simulation between the flower calculus and cut-free sequent calculus, yielding a soundness theorem and a *weak* completeness theorem for an analytic fragment of the flower calculus [16]. In this paper, we follow a *semantic* rather than syntactic approach, avoiding translations to and from symbolic formulas to obtain a stronger completeness result.

While currently at an early stage, we are also developing the *Flower Prover*, a prototype of direct-manipulation GUI for ITPs based on the flower calculus [17]. The interested reader can try a publicly available version of the prototype online[1]. We leave a detailed account of the Flower Prover and its connection to the flower calculus for future work.

Outline The article is organized as follows: in Section 2 we give a brief overview of the original diagrammatic syntax of EGs used by Peirce in his system Alpha for classical propositional logic. In Section 3 we retrace the origin of an intuitionistic variant of EGs first introduced by Oostra in [42], that directly inspired our flower metaphor. In Section 4 we illustrate quickly the original mechanism of lines of identity used by Peirce to handle first-order quantifiers in his Beta system, and show how to recast it in a more traditional binder-based syntax. In Section 5 we introduce our inductive syntax for flowers, and in Section 6 we give the full set of inference rules of the flower calculus as well as our notion of proof. In Section 7 we give a direct Kripke semantics to flowers, and in Section 8 we show that a restricted fragment of analytic and invertible rules is complete with respect to the semantics. Finally we conclude in Section 9 by a comparison with some related works.

▶ Note. For lack of space, we put the whole proof of soundness of the flower calculus in Appendix B. Contrary to the completeness proof, it is mostly routine work that does not require much insight. Detailed proofs for the deduction and completeness theorems are given respectively in Appendix C.1 and Appendix C.2. Readers already familiar with EGs can find a detailed comparison of the rules of the flower calculus with Peirce's illative transformations in Appendix A.

[1] `https://www.lix.polytechnique.fr/Labo/Pablo.DONATO/flowerprover/`

2 Existential graphs

Peirce designed in total three systems of EGs, which he called respectively Alpha, Beta and Gamma. They were invented chronologically in that order, which also captures their relationship in terms of complexity: Alpha is the foundation on which the other systems are built, and can today be understood as a diagrammatic calculus for classical *propositional* logic. As we will see in Section 4, Beta corresponds to a variable-free representation of *first-order* logic without function symbols. The last system Gamma is more experimental, with various unfinished features that have been interpreted as attempts to capture *modal* [59] and *higher-order* logics.

Sheet of Assertions The most fundamental concept of Alpha is the *sheet of assertion*, denoted by SA thereafter. It is the space where statements are scribed by the reasoner, typically a sheet of paper, a blackboard, or a computer display. As its name indicates, scribing a statement on SA amounts to *asserting its truth*. Thus naturally, the empty SA where nothing is scribed will denote *vacuous truth*, traditionally signified by the symbol $\top$.

Juxtaposition As we know from natural deduction, asserting the truth of the *conjunction* $a \wedge b$ of two propositions a and b, amounts to asserting *both* the truth of a and the truth of b. In Alpha, there is no need to introduce the symbolic connective $\wedge$, since one can just write both a and b at distinct locations on SA:

$$a \quad b$$

More generally, one might consider any two portions G and H of SA, and interpret their *juxtaposition* $G\ H$ as signifying that we assert the truth of their conjunction.

Cuts Asserting the truth of the *negation* $\neg a$ of a proposition a, amounts to *denying* the truth of a. This is done in Alpha by *enclosing* a in a closed curve like so:

$$\text{\textcircled{a}}$$

Peirce called such curves *cuts*[2], because they ought to be seen as literal cuts in the paper sheet that embodies SA. Note that they do not need to be circles: all that matters is that a is in a separate area from the rest of SA. This is precisely the content of the *Jordan curve theorem* in topology, and thus we can take cuts to be arbitrary Jordan curves. This entails in particular that cuts cannot intersect each other, but can be freely nested. Then as for juxtaposition, one can replace the proposition a in the interior of the cut by any *graph* G — i.e. any portion of SA — as long as the cut does not intersect other cuts in G.

Relationship with formulas With just these two *icons*, juxtaposition and cuts, one can therefore assert the truth of any proposition made up of conjunctions and negations and built from atomic propositions. Importantly, the only symbols needed for doing so are letters $a, b, c \ldots$ denoting atomic propositions, that is "pure" symbols that do not have any logical meaning associated to them.

Now, it is well-known that $\{\wedge, \neg\}$ is *functionally complete*, meaning that any boolean truth function can be expressed as the composition of conjunctions and negations. In particular,

[2] Not to be confused with the name given to instances of the *cut rule* in sequent calculus.

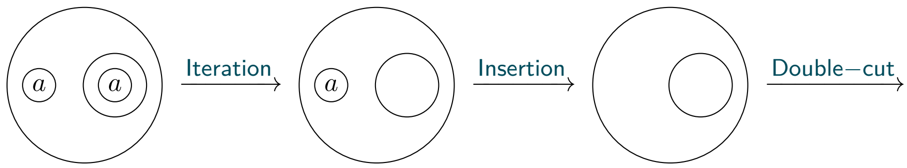
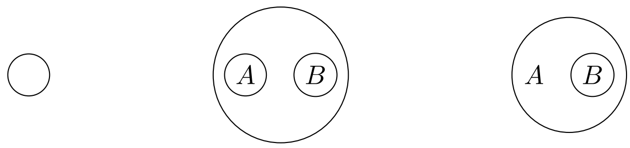


Figure 1 Proof of the law of excluded middle in Alpha

the symbolic definitions of *falsehood* $\bot \triangleq \neg\top$, classical *disjunction* $A \vee B \triangleq \neg(\neg A \wedge \neg B)$ and classical *implication* $A \supset B \triangleq \neg(A \wedge \neg B)$ can be expressed by the following three graphs:

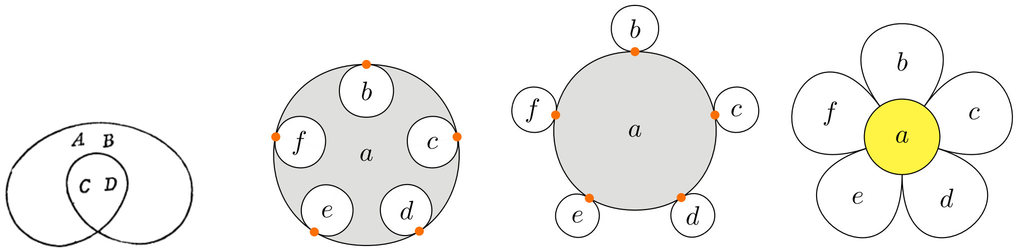


Thus one can easily encode any propositional formula into a classically equivalent graph. Conversely, one can translate any graph into a classically equivalent formula, as has been shown for instance in [50]. In fact, there are usually many possible formula readings of a given graph. One reason is that juxtaposition of graphs is a *variadic* operation, as opposed to conjunction of formulas which is *dyadic*: thus formulas that only differ up to *associativity* are associated to the same graph. Also, thanks to the topological nature of SA, juxtaposition is naturally *commutative*: the locations of two juxtaposed graphs do not matter, as long as they live in the same area delimited by a cut. The combination of these properties is called the *isotropy* of SA in [36], and is captured in traditional proof theory through the use of *(multi)sets* for modelling contexts in sequents.

Illative transformations In order to have a proof system, one needs a collection of *inference rules* for deducing true statements from other true statements. In Alpha, inference rules are implemented by what Peirce called *illative transformations* on graphs. In modern terminology, they correspond to *rewriting* rules that can be applied to any subgraph. By measuring the depth of a subgraph as the number of cuts in which it is enclosed, we thus have that the rules of Alpha are applicable on subgraphs of arbitrary depth. This makes Alpha deserving of the title of *deep inference* system.

Figure 1 shows a proof of the law of excluded middle $a \vee \neg a$ in Alpha. The first step consists in applying the illative transformation of Iteration to erase the subgraph a. More generally, Iteration allows to erase any subgraph G as long as G already occurs "higher" in SA, i.e. in an area that encloses the erased occurrence of G. The second step of Insertion allows to erase the other occurrence of a because it is scribed in a *negative* area, i.e. an area enclosed in an *odd* number of cuts — 1 in this case[3]. The last step of Double–cut allows to *collapse* the two remaining cuts because there is nothing but empty space in between them. This leaves us with the empty SA, having thus reduced the initial goal to trivial truth.

[3] It might be quite confusing that we call "Insertion" a transformation that *erases* information. This is because we use Peirce's original terminology, despite the fact that we adopt a *backward* reading of rules where the conclusion that we want to prove is reduced to a sufficient premiss.



(a) Peirce's scroll **(b)** Oostra's curl **(c)** Inside-out curl **(d)** Flower

Figure 2 From scrolls to flowers

3 Flowers

The scroll In [46, pp. 533–535], Peirce explains that he did not immediately come up with the idea of juxtaposition and cuts as diagrammatizations of conjunction and negation. Instead, they arose as the natural development of a more primitive icon that he called the *scroll*. Figure 2a shows Peirce's drawing of the scroll as it appears in [46, Fig. 5]. He defines its intended meaning as that of a "conditional de inesse", which corresponds to the material implication of classical logic. Then the graph of Figure 2a is interpreted as the formula $(A \wedge B) \supset (C \wedge D)$. This agrees with the encoding of implication given in Section 2, if one sees the outer boundary enclosing the antecedent $A\ B$ and the inner boundary enclosing the consequent $C\ D$ as nested cuts.

It is no coincidence that Peirce based his most fundamental icon on implication: according to Lewis [30, p. 79], he was the one who introduced the "illative relation" of implication into symbolic logic in the first place, by giving it a distinguished symbol and studying extensively the algebraic laws that govern it (e.g. Peirce's law $((A \supset B) \supset A) \supset A$).

The n-ary scroll In order to interpret the scroll as an *intuitionistic* implication, Oostra proposed in [42] to reify the scroll as a primitive icon of EGs, distinguished from the nesting of two cuts. In fact he went further, by generalizing both the cut and the scroll into an n-ary construction called the *curl*, where n is the number of inner boundaries, called *loops*. Figure 2b shows an example of curl with five loops, where the unique intersection points between inner and outer boundaries are highlighted in *orange*[4]. In [36], the curl is simply called *n-ary scroll*, the outer boundary *outloop*, and the inner boundaries *inloops*. Then cuts and scrolls are indeed special cases of n-ary scrolls, respectively with $n = 0$ and $n = 1$.

Like the unary scroll, the n-ary scroll is to be read as an implication whose antecedent is the content of the outloop, and consequent the content of the inloops. The generalization consists in taking the *disjunction* of the contents of all inloops: this reflects nicely the etymological meaning of the word "disjunction", since the inloops enclose *disjoint* areas of the outloop to which they are attached. Then the 5-ary scroll of Figure 2b can be read as the formula $a \supset (b \vee c \vee d \vee e \vee f)$; and the 0-ary scroll obtained by removing all inloops from the latter as $a \supset \bot$, since a 0-ary disjunction is naturally evaluated to its neutral element $\bot$. This coincides with the intuitionistic reading of negation $\neg A \triangleq A \supset \bot$.

[4] We also shade the negative area delimited by the outer boundary in *gray*.

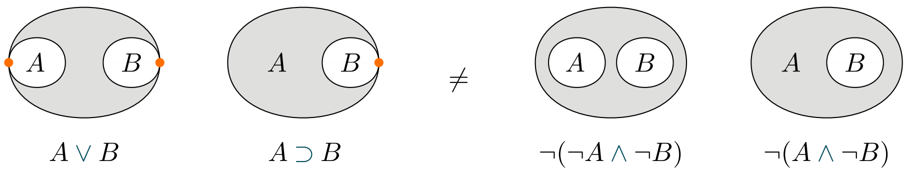


Figure 3 Continuity, disjunction and implication in intuitionistic EGs

Continuity With this interpretation of the n-ary scroll, the Alpha encodings of disjunction and implication as nested cuts given in Section 2 are no longer valid, because they are not intuitionistically equivalent to the associated binary and unary scrolls. This is illustrated in Figure 3, where the closeness in meaning is reflected iconically (but not symbolically) in the fact that the graphs only differ in the *continuity* (or lack thereof) between inloops and their outloop.

▶ Remark 3.1. This might be related to other manifestations of the notion of continuity in the semantics of intuitionistic logic, such as the well-known Stone-Tarski interpretation of formulas as topological spaces [53], and the interpretation of proofs as continuous maps in the *denotational semantics* of Dana Scott[5] [1].

Blooming In terms of ergonomy, the n-ary scroll has one notable flaw, also shared with the classical cut-based syntax: it quickly induces heavy nestings of curves in the plane, making even relatively simple graphs hard to read for an untrained eye. Our solution is to turn inloops *inside-out*, as illustrated in Figure 2c. In this way, we effectively divide the amount of curve-nesting in scrolls by two. And as an added bonus, the new icon is reminiscent of a *flower*, as if it had bloomed from its curled bud; or as if the pistol cylinder from Figure 2b had transformed into a *pistil*, and its bullet chambers into *petals*.

From that point onwards, we chose to fully embrace the flower metaphor: first in our drawing style as witnessed in Figure 2d, but also in our syntactic terminology, to be introduced in the next pages. Negative (resp. positive) outloops are now drawn as *yellow* (resp. *white*) pistils for a slightly more colorful experience, and inloops as transparent petals, i.e. of the same color as the area on which they are scribed.

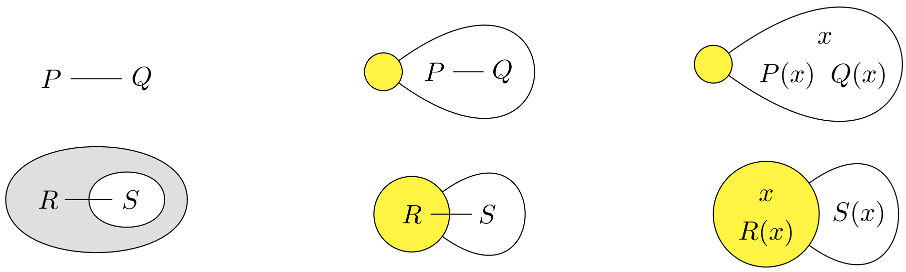
4 Gardens

Lines of identity To handle first-order quantification, Peirce introduced in Beta the icon of *lines of identity* (LoIs). In short, the usual binders and variables of predicate calculus are replaced by *lines* that connect the occurrences of bound variables in predicate arguments to their binding point. For instance, the formulas $\exists x.P(x) \wedge Q(x)$ and $\forall x.R(x) \supset S(x)$ can be represented in Beta by the graphs of Figure 4a.

The kind of quantification is determined by the location of the binding point, which is taken to be the *outermost* point in the line: if it is in a *positive* area as in the upper graph,

---

[5] Before the advent of Oostra's intuitionistic EGs, Zalamea gave a detailed analysis of Peirce's philosophy of the *continuum*, how it relates to modern developments in mathematics, and how it is embodied in EGs [58]. Actually according to Oostra [45, p. 162], "the possibility of developing intuitionistic existential graphs was first suggested by Zalamea in the 1990s [56, 57]".



(a) EGs with LoIs **(b)** Flowers with LoIs **(c)** Flowers with variables

Figure 4 From LoIs to variables

then the quantifier is *existential*; otherwise if it is in a *negative* area as in the lower graph, the quantifier is *universal.* This is justified by De Morgan's laws: the lower graph can also be read as the classically equivalent formula $\neg\exists x.R(x) \wedge \neg S(x)$.

Intuitionistic quantification In intuitionistic logic however, De Morgan's laws do not hold anymore. Thus in the flower calculus we need a different way to interpret LoIs as quantifiers. Our key insight is to adopt a *polarity-invariant* viewpoint: a LoI now has *existential* (resp. *universal*) force when its outermost point is located in a *petal* (resp. *pistil*). In particular, this implies that LoIs cannot occur at the top-level of SA anymore, but only inside flowers. Thus the two previous Beta graphs are transformed into the single-petal flowers of Figure 4b.

Variables Quine experimented with a notation similar to LoIs, but deemed it "too cumbersome for practical use" [48, p. 125]. While his lines connected locations inside symbolic formulas written in linear notation, it is true that having a line for each occurrence of bound variable can quickly lead to unreadable diagrams ridden with overlapping lines. This is not a problem in the context of Peirce's work, because his aim was "to separate [relational] reasoning into its smallest steps, [...] not to facilitate reasoning, but to facilitate the study of reasoning" [48, p. 111]; and recent formalizations of the algebra of LoIs in category theory support the pertinence of Peirce's approach [23, 6].

However, keeping in mind our goal of laying the basis for a calculus well-suited to practical reasoning in ITPs, we chose to replace LoIs by a more traditional syntax based on binders and variables. The idea is to substitute every LoI with a variable *binder* scribed in the area of its outermost point, so that the two flowers of Figure 4b transform into those of Figure 4c. Areas delimited by pistils and petals now comprise both flowers and binders, which can be seen metaphorically as *sprinklers* that irrigate the leaves (atomic predicates) of flowers through invisible LoIs, imagined as underground hoses. Hence we call these areas *gardens*.

5 Syntax

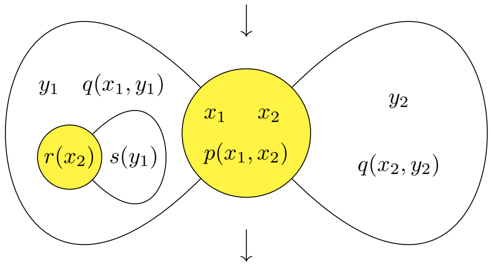
We are now going to distill the syntactic essence of flowers into an inductive, (multi)set-based data structure. This will allow for a more compact textual notation, that is better suited to proof-theoretical study. We previously illustrated how flowers allow to represent purely relational statements without function symbols. Since functions are just deterministic

Kind	Letters
Variables ($\mathcal{V}$)	x, y, z
Flowers ($\mathbb{F}$)	ϕ, ψ, ξ
Gardens ($\mathbb{G}$)	γ, δ
Sprinklers	$\mathbf{x}, \mathbf{y}, \mathbf{z}$
Variable vectors	$\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}$
Substitutions	σ, τ
Bouquets	Φ, Ψ, Ξ
Corollas	Γ, Δ
Contexts	$\hat{\Phi}, \hat{\Psi}, \hat{\Xi}$
Theories	$\mathcal{T}, \mathcal{U}$

(a) Conventions for meta-variables

$$x_1, x_2 \cdot p(x_1, x_2) \mathbin{\mathsf{D}} y_1 \cdot q(x_1, y_1), (r(x_2) \mathbin{\mathsf{D}} s(y_1))\, ;\, y_2 \cdot q(x_2, y_2)$$

$$\downarrow$$



$$\downarrow$$

$$\forall x_1 \forall x_2.(p(x_1, x_2) \supset (\exists y_1.q(x_1, y_1) \wedge (r(x_2) \supset s(y_1))) \vee (\exists y_2.q(x_2, y_2)))$$

(b) Interpreting flowers

Figure 5 Notations

relations, one can in principle formalize any first-order theory in this syntax[6].

▶ **Definition 5.1.** *A first-order signature is a pair $\Sigma = (\mathcal{P}, \mathsf{ar})$, where $\mathcal{P}$ is the countable set of predicate symbols of Σ, and $\mathsf{ar} : \mathcal{P} \to \mathbb{N}$ gives an arity to each symbol.*

In the following, we fix a countable set of variables $\mathcal{V}$ and a first-order signature Σ.

▶ **Definition 5.2.** *The sets of flowers $\mathbb{F}$ and gardens $\mathbb{G}$ are defined by mutual induction:*

Atom *If $p \in \mathcal{P}$ and $\vec{\mathbf{x}} \in \mathcal{V}^{\mathsf{ar}(p)}$, then $p(\vec{\mathbf{x}}) \in \mathbb{F}$;*
Garden *If $\mathbf{x} \subset \mathcal{V}$ is a finite set and $\Phi \subset \mathbb{F}$ a finite multiset, then $\mathbf{x} \cdot \Phi \in \mathbb{G}$;*
Flower *If $\gamma \in \mathbb{G}$ and $\Delta \subset \mathbb{G}$ is a finite multiset, then $\gamma \mathbin{\mathsf{D}} \Delta \in \mathbb{F}$.*

Building on our botanical metaphor, any finite set $\mathbf{x} \subset \mathcal{V}$ of variables is called a *sprinkler*, finite multiset $\Phi \subset \mathbb{F}$ of flowers a *bouquet*, and finite multiset $\Gamma \subset \mathbb{G}$ of gardens a *corolla*. We will often write gardens as $x_1, \ldots, x_n \cdot \phi_1, \ldots, \phi_m$, where the x_i are called *binders*; and non-atomic flowers as $\gamma \mathbin{\mathsf{D}} \delta_1 ; \ldots ; \delta_n$, where γ is the *pistil* and the δ_i are the *petals*. We write $\{E_i\}_i^n$ to denote a finite (multi)set of size n with elements E_i indexed by $1 \leq i \leq n$. We also omit writing the empty (multi)set, accounting for it with blank space as is done in sequent notation; in particular, $\cdot$ stands for the empty garden $\varnothing \cdot \varnothing$, $\gamma \mathbin{\mathsf{D}}$ for the flower with no petals $\gamma \mathbin{\mathsf{D}} \varnothing$, and $\gamma \mathbin{\mathsf{D}} \cdot$ for the flower with one empty petal.

Note that the order of precedence of operators is $, < \cdot < ; < \mathsf{D}$: this is illustrated in Figure 5b, where a flower expression is parsed into the corresponding flower drawing, and then translated as a formula. Also to improve readability, we will most of the time omit the garden dot '$\cdot$' when the sprinkler is empty, writing Φ instead of $\cdot\, \Phi$.

▶ Remark 5.3. In some places the choice of letter for meta-variables will be important to disambiguate the kind of syntactic object we denote. Table 5a summarizes our chosen notational conventions in this respect.

We now proceed with routine definitions for handling variables.

[6] Conversely, every relation can be faithfully encoded as its characteristic function, which is the basis for the formalization of mathematics in *type theories*.

▶ **Definition 5.4.** *The sets of* free variables $\mathsf{fv}(-)$ *and* bound variables $\mathsf{bv}(-)$ *of a flower/bouquet/garden are defined recursively by:*

$$\mathsf{fv}(p(\vec{\mathbf{x}})) = \vec{\mathbf{x}} \qquad \mathsf{fv}(\Phi) = \bigcup_{\phi \in \Phi} \mathsf{fv}(\phi) \qquad \mathsf{fv}(\mathbf{x} \cdot \Phi) = \mathsf{fv}(\Phi) \setminus \mathbf{x}$$

$$\mathsf{fv}(\mathbf{x} \cdot \Phi \mathrel{\mathsf{D}} \Delta) = \mathsf{fv}(\mathbf{x} \cdot \Phi) \cup \bigcup_{\mathbf{y} \cdot \Psi \in \Delta} \mathsf{fv}(\mathbf{x}, \mathbf{y} \cdot \Psi)$$

$$\mathsf{bv}(p(\vec{\mathbf{x}})) = \varnothing \quad \mathsf{bv}(\Phi) = \bigcup_{\phi \in \Phi} \mathsf{bv}(\phi) \quad \mathsf{bv}(\mathbf{x} \cdot \Phi) = \mathbf{x} \cup \mathsf{bv}(\Phi) \quad \mathsf{bv}(\gamma \mathrel{\mathsf{D}} \Delta) = \mathsf{bv}(\gamma) \cup \bigcup_{\delta \in \Delta} \mathsf{bv}(\delta)$$

To avoid reasoning about α-equivalence, we adopt in this work the so-called *Barendregt convention* that all variable binders are distinct, both among themselves and from free variables. Formally, we assume that for any bouquet Φ the two following conditions hold:

1. computing $\mathsf{bv}(\Phi)$ as a multiset gives the same result as computing it as a set;
2. $\mathsf{bv}(\Phi) \cap \mathsf{fv}(\Phi) = \varnothing$.

To define substitutions, we introduce a general notion of *function update*, which will be useful for the semantic evaluation of flowers in Section 7.

▶ **Definition 5.5.** *Let A, B be two sets, $f, g : A \to B$ two functions and $R \subseteq A$ some subset of their domain. The* update *of f on R with g is the function defined by:*

$$(f \,|_R\, g)(x) = \begin{cases} g(x) & \text{if } x \in R \\ f(x) & \text{otherwise} \end{cases}$$

$- \,|_{-}\, -$ *is left-associative, that is $f \,|_R\, g \,|_S\, h = (f \,|_R\, g) \,|_S\, h$. Also if f or g is the identity function 1 we omit writing it, i.e. $f \,|_R = f \,|_R\, 1$ and $|_R\, g = 1 \,|_R\, g$.*

▶ **Definition 5.6.** *A* substitution *is a function $\sigma : \mathcal{V} \to \mathcal{V}$ with a finite* support $\mathsf{supp}(\sigma) = \{x \mid \sigma(x) \neq x\}$. *We write $\sigma : \mathbf{x}$ to denote a substitution σ whose support is $\mathbf{x}$. The domain of substitutions is extended to flowers, bouquets and gardens mutually recursively by:*

$$\sigma(p(x_1, \ldots, x_n)) = p(\sigma(x_1), \ldots, \sigma(x_n)) \qquad \sigma(\phi_1, \ldots, \phi_n) = \sigma(\phi_1), \ldots, \sigma(\phi_n)$$

$$\sigma(\mathbf{x} \cdot \Phi) = \mathbf{x} \cdot \sigma|_{\mathbf{x}}(\Phi) \qquad \sigma(\mathbf{x} \cdot \Phi \mathrel{\mathsf{D}} \delta_1 ; \ldots ; \delta_n) = \sigma(\mathbf{x} \cdot \Phi) \mathrel{\mathsf{D}} \sigma|_{\mathbf{x}}(\delta_1) ; \ldots ; \sigma|_{\mathbf{x}}(\delta_n)$$

▶ **Definition 5.7.** *We say that a substitution $\sigma : \mathbf{x}$ is* capture-avoiding *in a bouquet Φ if $\sigma(\mathbf{x}) \cap \mathsf{bv}(\Phi) = \varnothing$.*

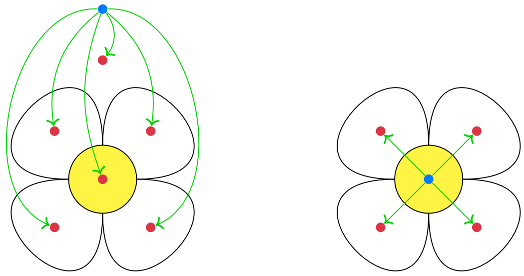
6 Calculus

Equipped with an inductive syntax, we can now express formally the inference rules of the flower calculus. First we need a notion of *context* to apply rules at arbitrarily deep locations:

▶ **Definition 6.1** (Context)**.** Contexts $\hat{\Phi}$ *are defined inductively by the following grammar:*

$$\hat{\Phi}, \hat{\Psi}, \hat{\Xi} ::= \Psi, \hat{\phi} \qquad \hat{\phi}, \hat{\psi}, \hat{\xi} ::= \square \mid \mathbf{x} \cdot \hat{\Phi} \mathrel{\mathsf{D}} \Delta \mid \gamma \mathrel{\mathsf{D}} \mathbf{x} \cdot \hat{\Phi} ; \Delta$$

Informally, a context can be seen as a bouquet with exactly one occurrence of a special flower $\square$ called its hole. *The* filling *of a context $\hat{\Phi}$ with a bouquet Ψ (resp. context $\hat{\Psi}$) is the bouquet $\hat{\Phi}\{\Psi\}$ (resp. context $\hat{\Phi}\{\hat{\Psi}\}$) where $\square$ has been subtituted with Ψ (resp. $\hat{\Psi}$).*



(a) Cross-pollination **(b)** Self-pollination

Figure 6 Pollination in flowers

▶ **Definition 6.2** (Polarity). *The number of* inversions $\mathsf{inv}(\hat{\Phi})$ *of a context* $\hat{\Phi}$ *is:*

$$\mathsf{inv}(\square) = 0 \qquad \mathsf{inv}(\Psi, \hat{\phi}) = \mathsf{inv}(\hat{\phi}) \qquad \mathsf{inv}(\mathbf{x} \cdot \hat{\Phi} \mathbin{\mathsf{D}} \Delta) = 1 + \mathsf{inv}(\hat{\Phi}) \qquad \mathsf{inv}(\gamma \mathbin{\mathsf{D}} \mathbf{x} \cdot \hat{\Phi} ; \Delta) = \mathsf{inv}(\hat{\Phi})$$

We say that a context $\hat{\Phi}$ *is* positive *if* $\mathsf{inv}(\hat{\Phi})$ *is even, and* negative *otherwise. We denote positive and negative contexts respectively by* $\hat{\Phi}^+$ *and* $\hat{\Phi}^-$*.*

In order to formulate the equivalent of the Iteration rule of EGs for flowers, we introduce a *pollination* relation that captures the availability of a flower in a given context:

▶ **Definition 6.3** (Pollination). *We say that a flower* ϕ *can be* pollinated *in a context* $\hat{\Phi}$*, written* $\phi \succ \hat{\Phi}$*, when there exists a bouquet* Ψ *with* $\phi \in \Psi$ *and contexts* $\hat{\Xi}$ *and* $\hat{\Xi}_0$ *s.t. either:*

Cross-pollination $\hat{\Phi} = \hat{\Xi}\{\Psi, \hat{\Xi}_0\}$*;*
Self-pollination $\hat{\Phi} = \hat{\Xi}\{\mathbf{x} \cdot \Psi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0 ; \Delta\}$ *for some* $\mathbf{x}, \mathbf{y}, \Delta$*.*

A bouquet Φ *can be* pollinated *in* $\hat{\Phi}$*, written* $\Phi \succ \hat{\Phi}$*, if* $\phi \succ \hat{\Phi}$ *for all* $\phi \in \Phi$*.*

Figure 6 illustrates the meaning of pollination as a relation of *justification* between locations: the blue dot marks the location of the justifying/pollinating occurrence of ϕ, and the red dots all the areas that it justifies/pollinates, and thus where ϕ is available for use. We distinguish two cases of cross-pollination and self-pollination, as botanists do when describing the reproduction of flowers. This distinction does not exist in classical EGs, because pistils and petals are both identified as instances of cuts[7].

▶ Remark 6.4. Incidentally, the pollination relation also explains the *scope* of variables. Indeed, one can interpet red dots in Figure 6 as the allowed *usage* points for the variable *bound* at the linked blue dot. This hints at a possible *type-theoretic* variant of the flower calculus where variables are also used for higher-order individuals, including flowers.

Proofs The inference rules of the flower calculus are presented in Figure 7. Read from top to bottom, they correspond to traditional inference rules deducing a necessary conclusion from a valid premiss. But we will prefer their backward, *bottom-up* reading: then they can be

[7] The same phenomenon is at work in SFL: cross-pollination and self-pollination can be seen as generalizing the *forward* and *backward* interaction connectives ○ and ▷ of intuitionistic SFL [13, 18], while the original formulation of SFL for classical linear logic had only one interaction connective $*$ [12]. This is also reminiscent of the adjunction between products (○) and exponentials (▷) in *cartesian closed categories*.

NATURE ❀

$$\frac{\hat{\Xi}\{\}}{\hat{\Xi}\{\Phi\}}\ \mathsf{poll}{\downarrow} \qquad \frac{\hat{\Xi}\{\Phi\}}{\hat{\Xi}\{\}}\ \mathsf{poll}{\uparrow}$$

$$\frac{\cdot\ \mathsf{D}\ \cdot\Phi}{\Phi}\ \mathsf{epis} \qquad \frac{}{\gamma\ \mathsf{D}\ \cdot\,;\Delta}\ \mathsf{epet}$$

$$\frac{\mathbf{x}\cdot\Phi\ \mathsf{D}\ \cdot\{\gamma_i\ \mathsf{D}\ \Delta\}_i^n}{\mathbf{x}\cdot\Phi,(\cdot\ \mathsf{D}\ \{\gamma_i\}_i^n)\ \mathsf{D}\ \Delta}\ \mathsf{srep}$$

$$\frac{(\mathbf{x}\cdot\sigma(\Phi)\ \mathsf{D}\ \sigma(\Delta)),(\mathbf{x},\mathbf{y}\cdot\Phi\ \mathsf{D}\ \Delta)}{\mathbf{x},\mathbf{y}\cdot\Phi\ \mathsf{D}\ \Delta}\ \mathsf{ipis}$$

$$\frac{\gamma\ \mathsf{D}\ \mathbf{x}\cdot\sigma(\Phi);\mathbf{x},\mathbf{y}\cdot\Phi;\Delta}{\gamma\ \mathsf{D}\ \mathbf{x},\mathbf{y}\cdot\Phi;\Delta}\ \mathsf{ipet}$$

CULTURE ✂

$$\frac{\hat{\Xi}^+\{\Phi\}}{\hat{\Xi}^+\{\}}\ \mathsf{grow} \qquad \frac{\hat{\Xi}^-\{\}}{\hat{\Xi}^-\{\Phi\}}\ \mathsf{crop}$$

$$\frac{\hat{\Xi}^+\{\gamma\ \mathsf{D}\ \Delta\}}{\hat{\Xi}^+\{\gamma\ \mathsf{D}\ \Gamma;\Delta\}}\ \mathsf{pull} \qquad \frac{\hat{\Xi}^-\{\gamma\ \mathsf{D}\ \Gamma;\Delta\}}{\hat{\Xi}^-\{\gamma\ \mathsf{D}\ \Delta\}}\ \mathsf{glue}$$

$$\frac{\hat{\Xi}^+\{\mathbf{x},\mathbf{y}\cdot\Phi\ \mathsf{D}\ \Delta\}}{\hat{\Xi}^+\{\mathbf{x}\cdot\sigma(\Phi)\ \mathsf{D}\ \sigma(\Delta)\}}\ \mathsf{apis}$$

$$\frac{\hat{\Xi}^-\{\gamma\ \mathsf{D}\ \mathbf{x},\mathbf{y}\cdot\Phi;\Delta\}}{\hat{\Xi}^-\{\gamma\ \mathsf{D}\ \mathbf{x}\cdot\sigma(\Phi);\Delta\}}\ \mathsf{apet}$$

In the rules poll↓ and poll↑, we assume that $\Phi \succ \hat{\Xi}$.

In the rules ipis, apis (resp. ipet, apet), we assume some substitution $\sigma : \mathbf{y}$ that is capture-avoiding in $\cdot\ \Phi\ \mathsf{D}\ \Delta$ (resp. Φ).

Figure 7 Rules of the flower calculus

seen as *rewriting* rules that reduce a goal to a sufficient premiss, just like in our illustration of the illative transformations of EGs in Figure 1. Also, all rules manipulate *bouquets*: this is seen more clearly in the *graphical* presentation of the rules in appendix (Figures 8 and 9).

We partition the rules into two sets: the *natural* rules denoted by ❀ that apply in arbitrary contexts, and the *cultural* rules denoted by ✂ that apply exclusively in positive or negative contexts. In particular, every ❀-rule is both *analytic* (i.e. every atom in the premiss already appears in the conclusion) and *invertible* (Lemma B.17); on the contrary, all ✂-rules are *non-invertible*, and they will be shown to be *admissible* in Section 8.

▶ **Definition 6.5** (Derivation). *Given a set of rules* R*, we write* $\Phi \to_{\mathsf{R}} \Psi$ *to indicate a rewrite* step *in* R*, that is an instance of some* $\mathsf{r} \in \mathsf{R}$ *with* Ψ *as premiss and* Φ *as conclusion. We just write* $\Phi \to \Psi$ *to mean* $\Phi \to_{❀\cup✂} \Psi$*. A* derivation $\Phi \to^n_{\mathsf{R}} \Psi$ *is a sequence of rewrite steps* $\Phi_0 \to_{\mathsf{R}} \Phi_1 \ldots \to_{\mathsf{R}} \Phi_n$ *with* $\Phi_0 = \Phi$*,* $\Phi_n = \Psi$ *and* $n \geq 0$*. Generally the length* n *of the derivation does not matter, and we just write* $\Phi \to^*_{\mathsf{R}} \Psi$*. Finally, natural derivations are closed under arbitrary contexts: for every context* $\hat{\Xi}$*,* $\Phi \to_{❀} \Psi$ *implies* $\hat{\Xi}\{\Phi\} \to_{❀} \hat{\Xi}\{\Psi\}$*. We write* $\Phi \rightharpoonup_{❀} \Psi$ *to denote a* shallow *natural step, i.e. an instance of a ❀-rule in the empty context* $\square$*.*

▶ **Definition 6.6** (Proof). *A* proof *of a bouquet* Φ *is a derivation* $\Phi \to^* \varnothing$*.*

In Peircean terms, the empty bouquet is the blank SA. Then proving a bouquet amounts to erasing it completely from SA, thus reducing it to trivial truth as in Figure 1. Figure 10 shows an example of ❀-proof in the flower calculus, both in textual and graphical syntax. Note that we used a non-duplicating version of the rules ipis and ipet, in order to save some space in the graphical presentation.

If we want to speak about *relative* truth, i.e. Φ is true under the assumption that Ψ is, we can simply rely on the existence of a derivation $\Phi \rightarrow^* \Psi$ in the full flower calculus. This will be justified by the soundness of all rules (Theorem B.20) as well as a *strong* completeness result (Corollary 8.8), that relies on the following strong deduction theorem:

▶ **Theorem 6.7** (Strong deduction). *$\Phi \rightarrow^* \Psi$ if and only if $\Psi \supset \Phi \rightarrow^* \varnothing$.*

Contrary to full derivability, natural derivability $\Phi \rightarrow^*_{❀} \Psi$ is too weak to satisfy a strong deduction theorem. This is a consequence of the fact that ❀-rules are *invertible*, and thus can only relate equivalent bouquets. Indeed, as soon as $\Psi \supset \Phi$ is ❀-provable but the converse $\Phi \supset \Psi$ is not, it follows from the completeness of ❀-rules that Φ and Ψ are not equivalent: thus $\Phi \not\rightarrow^*_{❀} \Psi$, contradicting the strong deduction statement.

A trivial way to circumvent this is to define directly $\Psi \vdash \Phi$ as $\Psi \supset \Phi \rightarrow^* \varnothing$. This is closer to what one would find in sequent calculus, where hypothetical proofs are closed derivations of hypothetical sequents, not open derivations. The difference is that sequents capture only the *first-order*[8] implicative structure of logic, while flowers capture the full structure of intuitionistic FOL. This allows for a nice generalization of the notion of *hypothetical provability*, which will be useful in our completeness proof:

▶ **Definition 6.8.** *We say that Φ is hypothetically provable from Ψ in a fragment R of rules, written $\Psi \vdash_{\mathsf{R}} \Phi$, if $\hat{\Xi}\{\Phi\} \rightarrow^*_{\mathsf{R}} \hat{\Xi}\{\}$ for every context $\hat{\Xi}$ such that $\Psi \succ \hat{\Xi}$. We write $\Psi \vdash \Phi$ to denote hypothetical provability in the full flower calculus.*

▶ **Theorem 6.9** (Deduction). *$\Psi \vdash_{❀} \Phi$ if and only if $\vdash_{❀} \Psi \supset \Phi$.*

7 Semantics

We now give a semantics to flowers in Kripke structures. We recall the standard definitions:

▶ **Definition 7.1.** *A first-order structure is a pair $(M, [\![\cdot]\!])$ where M is a non-empty set called the domain, and $[\![\cdot]\!]$ is a map called the interpretation that associates to each predicate symbol $p \in \mathcal{P}$ a relation $[\![p]\!] \subseteq M^{\mathsf{ar}(p)}$.*

▶ **Definition 7.2.** *A Kripke structure is a triplet $\mathcal{K} = (W, \leq, (M_w)_{w \in W})$, where W is the set of worlds, $\leq$ is a pre-order on W called accessibility, and $(M_w)_{w \in W}$ is a family of first-order structures indexed by W. Furthermore, we require the following monotonicity conditions to hold whenever $w \leq w'$:* **1.** *$M_w \subseteq M_{w'}$;* **2.** *for every $p \in \mathcal{P}$, $[\![p]\!]_w \subseteq [\![p]\!]_{w'}$.*

▶ **Definition 7.3.** *Given a Kripke structure $\mathcal{K}$ and a world w in $\mathcal{K}$, a w-evaluation is a function $e : \mathcal{V} \rightarrow M_w$. The interpretation map of M_w is extended to variables and substitutions with respect to any w-evaluation e as follows:*

$$[\![x]\!]_e = e(x) \qquad\qquad [\![\sigma]\!]_e(x) = [\![\sigma(x)]\!]_e$$

The crux of Kripke semantics is the *forcing* relation, that captures the truth-conditions of statements in Kripke structures. While it is usually defined on formulas, here we adapt the definition to flowers, which in our opinion makes it simpler and more uniform since flowers can be seen as built from essentially one big constructor:

[8] As opposed to *higher-order*, in the sense of having negatively nested implications.

▶ **Definition 7.4.** *The depth $|-|$ of a flower/garden is defined by mutual recursion:*

$$|p(\vec{\mathbf{x}})| = 0 \qquad |\mathbf{x} \cdot \Phi| = \max_{\phi \in \Phi} |\phi| \qquad |\gamma \mathbin{\supset} \Delta| = 1 + \max(|\gamma|, \max_{\delta \in \Delta} |\delta|)$$

▶ **Definition 7.5.** *Given some Kripke structure $\mathcal{K}$, the forcing relation $w \Vdash \phi\,[e]$ between a world w, a flower ϕ and a w-evaluation e is defined by induction on $|\phi|$ as follows:*

Atom $w \Vdash p(\vec{\mathbf{x}})\,[e]$ *iff* $[\![\vec{\mathbf{x}}]\!]_e \in [\![p]\!]_w$*;*
Flower $w \Vdash \mathbf{x} \cdot \Phi \mathbin{\supset} \{\mathbf{x}_i \cdot \Phi_i\}_i^n\,[e]$ *iff for every $w' \geq w$ and every w'-evaluation e', if $w' \Vdash \Phi\,[e|_{\mathbf{x}} e']$ then there is some $1 \leq i \leq n$ and w'-evaluation e'' such that $w' \Vdash \Phi_i\,[e|_{\mathbf{x}} e'|_{\mathbf{x}_i} e'']$.*
Bouquet $w \Vdash \Phi\,[e]$ *iff* $w \Vdash \phi\,[e]$ *for every* $\phi \in \Phi$*.*

Lastly, we define the notion of *semantic entailment* $\Phi \vDash \Psi$ on bouquets, mirroring the syntactic entailment $\Phi \vdash \Psi$ of the last section:

▶ **Definition 7.6.** *Let $\mathcal{K}$ be a Kripke structure, and Φ, Ψ some bouquets. We say that Φ semantically entails Ψ in $\mathcal{K}$, written $\Phi \vDash_{\mathcal{K}} \Psi$, when $w \Vdash \Phi\,[e]$ implies $w \Vdash \Psi\,[e]$ for every world $w \in W$ and w-evaluation e. This entailment is valid if it holds for any Kripke structure $\mathcal{K}$, and in that case we simply write $\Phi \vDash \Psi$. We say that Φ is semantically equivalent to Ψ, written $\Phi \mathrel{=\!\!\!|\!\vDash} \Psi$, when $\Phi \vDash \Psi$ and $\Psi \vDash \Phi$.*

8 Completeness

We now outline a direct completeness proof for the natural fragment ❀ of the flower calculus: every true flower ϕ is naturally provable, i.e. $\vDash \phi$ implies $\vdash_{❀} \phi$. Since this fragment is analytic, we cannot reuse most completeness proofs from the literature, because they usually rely on a non-analytic principle like the cut rule of sequent calculus. Our insight was to adapt techniques from the *semantic cut-elimination* proof given by Hermant in [24], which is nonetheless relatively close to the original completeness proof of Gödel. A novelty of our proof is that it dispenses completely with the need for *Henkin witnesses*.

First we need to generalize our notions of syntactic and semantic entailment to possibly *infinite* sets of flowers, so-called *theories*:

▶ **Definition 8.1.** *Any set $\mathcal{T} \subseteq \mathbb{F}$ of flowers is called a theory. In particular, a bouquet can be regarded as a finite theory, by forgetting the number of repetitions of its elements. We say that a bouquet Φ is* provable *from a theory $\mathcal{T}$, written $\mathcal{T} \vdash \Phi$, if there exists a bouquet $\Psi \subseteq \mathcal{T}$ such that $\Psi \vdash \Phi$. Given a Kripke structure $\mathcal{K}$, a world w in $\mathcal{K}$ and a w-evaluation e, we say that $\mathcal{T}$ is forced by w under e, written $w \Vdash \mathcal{T}\,[e]$, if $w \Vdash \phi\,[e]$ for all $\phi \in \mathcal{T}$. Then Φ is a consequence of $\mathcal{T}$, written $\mathcal{T} \vDash_{\mathcal{K}} \Phi$, if $w \Vdash \mathcal{T}\,[e]$ implies $w \Vdash \Phi\,[e]$ for every world w in $\mathcal{K}$ and w-evaluation e.*

▶ **Definition 8.2.** *A theory $\mathcal{T}$ is said to be ψ-consistent when $\mathcal{T} \nvdash_{❀} \psi$, and ψ-complete when for all $\phi \in \mathbb{F}$, either $\mathcal{T}, \phi \vdash_{❀} \psi$ or $\phi \in \mathcal{T}$.*

The following two propositions constitute the central argument that allows the completeness proof to go through despite the analyticity of ❀-rules. They are a direct adaptation of [24, Proposition 7], which Hermant identifies as “an important property of any A-consistent, A-complete theory, [...] that it enjoys some form of the subformula property”.

▶ **Proposition 8.3** (Analytic truth)**.** *Let $\psi \in \mathbb{F}$, $\mathcal{T}$ some ψ-consistent and ψ-complete theory, and $\phi = \mathbf{x} \cdot \Phi \mathbin{\supset} \Delta$ with $\Delta = \{\delta_i\}_i^n = \{\mathbf{x}_i \cdot \Phi_i\}_i^n$ such that $\phi \in \mathcal{T}$. Then for every substitution $\sigma : \mathbf{x}$, either $\sigma(\Phi_i) \subseteq \mathcal{T}$ for some $1 \leq i \leq n$, or $\mathcal{T} \nvdash_{❀} \sigma(\Phi)$.*

▶ **Proposition 8.4** (Analytic refutation). *Let $\psi \in \mathbb{F}$, $\mathcal{T}$ some ψ-consistent and ψ-complete theory, and $\phi = \mathbf{x} \cdot \Phi \supset \Delta$ with $\Delta = \{\delta_i\}_i^n = \{\mathbf{x}_i \cdot \Phi_i\}_i^n$ such that $\mathcal{T} \nvdash_{❀} \phi$. Then for every $1 \leq i \leq n$ and substitution $\sigma : \mathbf{x}_i$, there is some $\phi_i \in \Phi_i$ such that $\mathcal{T}, \Phi \nvdash_{❀} \sigma(\phi_i)$.*

Next, we define the so-called *universal Kripke structure* ✿(ψ) relative to a flower ψ:

▶ **Definition 8.5.** *Let $\psi \in \mathbb{F}$. The universal Kripke structure ✿(ψ) has:*

- *The set of ψ-consistent and ψ-complete theories as its worlds;*
- *Set inclusion $\subseteq$ as its accessibility relation;*
- *For each world $\mathcal{T}$, a first-order structure whose domain is the set of variables $\mathcal{V}$, and whose interpretation map is given by $[\![p]\!]_{\mathcal{T}} = \{\vec{\mathbf{x}} \mid p(\vec{\mathbf{x}}) \in \mathcal{T}\}$.*

One can easily check that the monotonicity conditions of Kripke structures hold for ✿(ψ).

We are now equipped to formulate the main *adequacy* lemma, which relates forcing in ✿(ψ) with ψ-consistency and ψ-completeness thanks to Propositions 8.3 and 8.4:

▶ **Lemma 8.6** (Adequacy). *Let $\phi, \psi \in \mathbb{F}$, $\mathcal{T}$ a ψ-consistent and ψ-complete theory, and σ a substitution. Then* **1.** *$\sigma(\phi) \in \mathcal{T}$ implies $\mathcal{T} \Vdash \phi\,[\sigma]$, and* **2.** *$\mathcal{T} \nvdash_{❀} \sigma(\phi)$ implies $\mathcal{T} \nVdash \phi\,[\sigma]$.*

As a near-direct consequence, we get:

▶ **Theorem 8.7** (Completeness). *$\Phi \vDash \Psi$ implies $\Phi \vdash_{❀} \Psi$.*

Combined with strong deduction (Theorem 6.7), this also yields a strong completeness theorem for the full flower calculus[9]:

▶ **Corollary 8.8** (Strong completeness). *$\Phi \vDash \Psi$ implies $\Psi \to^* \Phi$.*

Finally, the composition of the soundness, completeness and deduction theorems (B.20, 8.7 and 6.9) gives the admissibility of ✂-rules, and thus the analyticity of the flower calculus:

▶ **Corollary 8.9** (Analyticity). *If $\Phi \vdash \Psi$ then $\Phi \vdash_{❀} \Psi$.*

9 Related works

Intuitionistic EGs We have already mentioned the seminal work of Oostra, who introduced in [42] an intuitionistic version of Alpha. In [43] he describes its natural extension with LoIs to get an intuitionistic version of Beta, and in [44] he gives formal soundness and completeness proofs for intuitionistic Alpha, based on a linear notation for graphs. Ma and Pietarinen have developed in [36] their own system of intuitionistic EGs for propositional logic, with a different set of inference rules than Oostra's. They give a more systematic proof theory, including deduction, soundness and completeness theorems with respect to Heyting algebras.

Our work brings several new contributions on top of those:

Variadicity Our multiset-based definition of flowers captures faithfully the *variadic* nature of juxtaposition and n-ary scrolls in the diagrammatic syntax. In contrast, previous formalizations rely on a restricted inductive syntax which only captures graphs that are isomorphic to formulas built with binary connectives.

[9] Actually it already works for the fragment ❀ $\cup$ {grow}, thanks to the proof of the strong deduction theorem (see Appendix C.1).

Intuitionistic binders While replacing LoIs with binders and variables has already been done by Sowa in the context of classical EGs [52], it seems like we are the first to adapt the idea to the intuitionistic setting.

Analyticity To our knowledge, we are the first to give a Kripke semantics to a syntax based on EGs, and to use this to obtain an analyticity result[10].

Invertibility The natural fragment of the flower calculus appears to be the first proof system based on EGs where all rules are *invertible*.

Categorical EGs Since the seminal work of Brady and Trimble in 2000 on the formalization of EGs in category theory [7, 8], there have been various efforts to find rich categorical axiomatizations of Beta. The first approach — initiated in [8] — is based on *string diagrams*, and has recently enabled strong connections with *Frobenius algebras* and *bicategories* [39, 23, 6]. A second approach makes use of the concept of *generic figure* [11], introduced by Reyes as a basic building block for *topos theory* [29]. We do not know however of any attempt to uncover the categorical structures underlying intuitionistic EGs. The flower calculus might be an interesting candidate, in that the invertibility of the natural fragment could enable a purely *equational* approach.

Deep inference While the deep inference literature is most furnished with systems for classical logic, a few works tackle intuitionistic logics: the seminal work of Tiu, who proposed a *calculus of structures* for intuitionistic FOL [55], was followed by computational interpretations of the implicational fragment in Guenot's thesis [22]. There are also *nested sequent* systems for (propositional) full intuitionistic linear logic [15], standard and constant-domain intuitionistic FOL [19], and intuitionistic modal logics [14, 28, 33]. The flower calculus is closer to Guenot's nested sequent calculi which also work as rewriting systems, but generalizes them to full intuitionistic FOL.

Labelled sequent calculi For a long time, it was believed that there could not be fully invertible proof systems for intuitionistic logics, even in the propositional case. While this might be true in standard Gentzen formalisms, recent works have shown that it is possible in the context of *labelled sequent calculi*: first with Lyon's G3IntQ calculus for FOL [32, Section 3.3], and then with the calculus $\mathsf{labIS4}_{\leq}$ of Girlando et al. for the modal logic S4 [21]. In these systems, invertibility is made possible by the addition of *semantic* information to sequents, in the form of so-called *labels* and *relational atoms* that respectively encode the worlds and accessibility relations of Kripke structures. The flower calculus follows instead a purely *syntactic* approach, by relying on deep inference to retrieve what would normally be semantic information from the context $\hat{\Xi}$ in the pollination rules $\mathsf{poll}{\uparrow}$ and $\mathsf{poll}{\downarrow}$.

Coherent logic We noticed a formal connection between flowers and *coherent logic*, a subset of the formulas of FOL discovered by Skolem in 1920 [51] that is capable of expressing many mathematical theories, and has close connections to topos theory [27, Section D3.3]. Indeed,

[10] Ma and Pietarinen claim in [35] that Alpha is analytic because it can simulate the cut rule of sequent calculus. This is a misinterpretation, since this supports precisely the *contrary*: the ability to simulate the cut rule with a constant number of rules implies the *non-analyticity* of one the rules involved (namely, Peirce's Deletion rule). Still, the notion of analyticity is not yet fully understood in deep inference systems, as discussed in [10].

the interpretation $\lfloor \mathbf{x} \cdot \Phi \rhd \Delta \rfloor$ of a generic flower is given by the following formula, which has exactly the shape of a coherent formula as described e.g. in [5]:

$$\forall \mathbf{x}. \left(\bigwedge_{\phi \in \Phi} \lfloor \phi \rfloor \supset \bigvee_{\mathbf{y} \cdot \Psi \in \Delta} \exists \mathbf{y}. \bigwedge_{\psi \in \Psi} \lfloor \psi \rfloor \right)$$

The only difference is that flowers can be *nested*, while coherent formulas (also called coherent *sequents*) are first-order, in the sense that ϕ and ψ must be atoms. Coherent formulas appear in the theory of *focusing* in sequent calculi [37], and they lend themselves to simple proof search procedures that allow for *explainable proof automation* in ITPs [5, 26]. A higher-order variant of coherent formulas that is almost isomorphic to flowers has also been used to construct an intuitionistic version of the *arithmetical hierarchy*, as well as a fully *non-invertible* proof system for propositional intuitionistic logic [9].

Development calculi Through their backward reading, the rules of the flower calculus can be understood as primitive *tactics* for building proofs interactively. In [2, Chapter 3], Ayers calls such systems *development calculi*. In particular, he presents his own development calculus inspired by McBride's OLEG system [38] and Ganesalingam & Gowers's prover [20] called the `Box` calculus, where goals are represented by a so-called `Box` data structure very similar to flowers. In particular, `Box`es have so-called *disjunctive pairs* to reduce backtracking, that correspond to the petals of flowers. The main difference is that the `Box` calculus is based on dependent type theory instead of FOL: this allows to store the partial proof terms inside of the `Box`es themselves, while this information is lost during the construction of flowers. However, there is no completeness nor analyticity result for the `Box` calculus. It would be interesting to investigate further connections, in order to develop a dependently-typed version of the flower calculus.

References

1. Samson Abramsky and Achim Jung. *Domain Theory*. Oxford University Press, Inc., USA, 1995.
2. Edward W. Ayers. *A Tool for Producing Verified, Explainable Proofs*. PhD thesis, University of Cambridge, 2021.
3. Edward W. Ayers, Mateja Jamnik, and W. T. Gowers. A Graphical User Interface Framework for Formal Verification. In Liron Cohen and Cezary Kaliszyk, editors, *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITP.2021.4`, `doi:10.4230/LIPIcs.ITP.2021.4`.
4. Yves Bertot, Gilles Kahn, and Laurent Théry. Proof by pointing. In Masami Hagiya and John C. Mitchell, editors, *Theoretical Aspects of Computer Software*, volume 789, pages 141–160. Springer Berlin Heidelberg, 1994. Series Title: Lecture Notes in Computer Science. URL: `http://link.springer.com/10.1007/3-540-57887-0_94`, `doi:10.1007/3-540-57887-0_94`.
5. Marc Bezem and Thierry Coquand. Automating Coherent Logic. In Geoff Sutcliffe and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, Lecture Notes in Computer Science, pages 246–260, Berlin, Heidelberg, 2005. Springer. `doi:10.1007/11591191_18`.
6. Filippo Bonchi, Alessandro Di Giorgio, Nathan Haydon, and Pawel Sobocinski. Diagrammatic Algebra of First Order Logic, January 2024. arXiv:2401.07055 [cs, math]. URL: `http://arxiv.org/abs/2401.07055`, `doi:10.48550/arXiv.2401.07055`.

7 Geraldine Brady and Todd H. Trimble. A categorical interpretation of C.S. Peirce's propositional logic Alpha. *Journal of Pure and Applied Algebra*, 149(3):213–239, June 2000. URL: https://linkinghub.elsevier.com/retrieve/pii/S0022404998001790, doi:10.1016/S0022-4049(98)00179-0.

8 Geraldine Brady and Todd H. Trimble. A String Diagram Calculus for Predicate Logic and C. S. Peirce's System Beta, June 2000. URL: https://ncatlab.org/nlab/files/BradyTrimbleString.pdf.

9 Taus Brock-Nannestad and Danko Ilik. An Intuitionistic Formula Hierarchy Based on High-School Identities. *Mathematical Logic Quarterly*, 65(1):57–79, May 2019. arXiv: 1601.04876. URL: http://arxiv.org/abs/1601.04876, doi:10.1002/malq.201700047.

10 Paola Bruscoli and Alessio Guglielmi. On Analyticity in Deep Inference. *Mathematical Structures in Computer Science*, 29(Special Issue 8: A special issue on structural proof theory, automated reasoning and computation in celebration of Dale Miller's 60th birthday), 2019. URL: https://people.bath.ac.uk/ag248/p/ADI.pdf, doi:10.1017/S0960129519000136.

11 Gianluca Caterina and Rocco Gangle. A New Syntax for Diagrammatic Logic: A Generic Figures Approach. In Yaroslav D. Sergeyev and Dmitri E. Kvasov, editors, *Numerical Computations: Theory and Algorithms*, Lecture Notes in Computer Science, pages 43–58, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-40616-5_4.

12 Kaustuv Chaudhuri. Subformula linking as an interaction method. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving*, volume 7998, pages 386–401. Springer Berlin Heidelberg, 2013. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/978-3-642-39634-2_28, doi:10.1007/978-3-642-39634-2_28.

13 Kaustuv Chaudhuri. Subformula linking for intuitionistic logic with application to type theory. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 200–216. Springer, 2021. doi:10.1007/978-3-030-79876-5_12.

14 Kaustuv Chaudhuri, Sonia Marin, and Lutz Straßburger. Modular focused proof systems for intuitionistic modal logics. In *International Conference on Formal Structures for Computation and Deduction*, 2016.

15 Ranald Clouston, Jeremy Dawson, Rajeev Goré, and Alwen Tiu. Annotation-Free Sequent Calculi for Full Intuitionistic Linear Logic. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 197–214, Dagstuhl, Germany, 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2013.197, doi:10.4230/LIPIcs.CSL.2013.197.

16 Pablo Donato. flowers-metatheory, November 2022. URL: https://github.com/Champitoad/flowers-metatheory.

17 Pablo Donato. flower-prover, October 2023. URL: https://github.com/Champitoad/flower-prover.

18 Pablo Donato, Pierre-Yves Strub, and Benjamin Werner. A drag-and-drop proof tactic. In *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2022, page 197–209, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3497775.3503692.

19 Melvin Fitting. Nested Sequents for Intuitionistic Logics. *Notre Dame Journal of Formal Logic*, 55(1):41–61, 2014. URL: http://projecteuclid.org/euclid.ndjfl/1390246437, doi:10.1215/00294527-2377869.

20 M. Ganesalingam and W. T. Gowers. A Fully Automatic Theorem Prover with Human-Style Output. *Journal of Automated Reasoning*, 58(2):253–291, February 2017. doi:10.1007/s10817-016-9377-1.

21 Marianna Girlando, Roman Kuznets, Sonia Marin, Marianela Morales, and Lutz Straßburger. Intuitionistic S4 is decidable. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, June 2023. arXiv:2304.12094 [cs]. URL: http://arxiv.org/abs/2304.12094, doi:10.1109/LICS56636.2023.10175684.

22 Nicolas Guenot. *Nested Deduction in Logical Foundations for Computation*. PhD thesis, Ecole Polytechnique X, April 2013. URL: https://pastel.archives-ouvertes.fr/pastel-00929908.

23 Nathan Haydon and Paweł Sobociński. Compositional Diagrammatic First-Order Logic. In Ahti-Veikko Pietarinen, Peter Chapman, Leonie Bosveld-de Smet, Valeria Giardino, James Corter, and Sven Linker, editors, *Diagrammatic Representation and Inference*, volume 12169, pages 402–418. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/978-3-030-54249-8_32, doi:10.1007/978-3-030-54249-8_32.

24 Olivier Hermant. Semantic Cut Elimination in the Intuitionistic Sequent Calculus. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, and Paweł Urzyczyn, editors, *Typed Lambda Calculi and Applications*, volume 3461, pages 221–233. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science. URL: http://link.springer.com/10.1007/11417170_17, doi:10.1007/11417170_17.

25 John Howse, Gem Stapleton, and John Taylor. Spider Diagrams. *LMS Journal of Computation and Mathematics*, 8:145–194, January 2005. Publisher: Cambridge University Press. URL: https://www.cambridge.org/core/journals/lms-journal-of-computation-and-mathematics/article/spider-diagrams/B312DE2E402F39F9BEF75770F80559D0#, doi:10.1112/S1461157000000942.

26 Predrag Janičić and Julien Narboux. Automated generation of illustrated proofs in geometry and beyond. *Annals of Mathematics and Artificial Intelligence*, July 2023. URL: https://link.springer.com/10.1007/s10472-023-09857-y, doi:10.1007/s10472-023-09857-y.

27 Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, volume 2 of *Oxford Logic Guides*. Clarendon Press, Oxford, England, September 2002.

28 Roman Kuznets and Lutz Straßburger. Maehara-style modal nested calculi. *Archive for Mathematical Logic*, 58(3-4):359–385, May 2019. URL: http://link.springer.com/10.1007/s00153-018-0636-1, doi:10.1007/s00153-018-0636-1.

29 Marie La Palme Reyes, Gonzalo E. Reyes, and Houman Zolfaghari. *Generic figures and their glueings*. Polimetrica, International Scientific Publisher, 2008.

30 C. I. Lewis. A survey of symbolic logic. *Journal of Philosophy, Psychology and Scientific Methods*, 17(3):78–79, 1920. doi:10.2307/2940631.

31 Sven Linker, Jim Burton, and Mateja Jamnik. Tactical Diagrammatic Reasoning. *Electronic Proceedings in Theoretical Computer Science*, 239:29–42, January 2017. URL: http://arxiv.org/abs/1701.07126, doi:10.4204/EPTCS.239.3.

32 Tim Lyon. *Refining Labelled Systems for Modal and Constructive Logics with Applications*. PhD thesis, July 2021. URL: http://arxiv.org/abs/2107.14487, doi:10.48550/arXiv.2107.14487.

33 Tim S. Lyon. Nested Sequents for Intuitionistic Modal Logics via Structural Refinement. In Anupam Das and Sara Negri, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, Lecture Notes in Computer Science, pages 409–427, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-86059-2_24.

34 Thomas Långbacka, Rimvydas Rukšėnas, and Joakim von Wright. TkWinHOL: A tool for Window Inference in HOL. In E. Thomas Schubert, Philip J. Windley, and James Alves-Foss, editors, *Higher Order Logic Theorem Proving and Its Applications*, Lecture Notes in Computer Science, pages 245–260, Berlin, Heidelberg, 1995. Springer. doi:10.1007/3-540-60275-5_69.

35 Minghui Ma and Ahti-Veikko Pietarinen. Proof Analysis of Peirce's Alpha System of Graphs. *Studia Logica*, 105(3):625–647, June 2017. URL: http://link.springer.com/10.1007/s11225-016-9703-y, doi:10.1007/s11225-016-9703-y.

36 Minghui Ma and Ahti-Veikko Pietarinen. A graphical deep inference system for intuitionistic logic. *Logique et Analyse*, 245:73–114, January 2019. URL: http://www.scopus.com/inward/record.url?scp=85066258215&partnerID=8YFLogxK, doi:10.2143/LEA.245.0.3285706.

37 Sonia Marin, Dale Miller, Elaine Pimentel, and Marco Volpe. From axioms to synthetic inference rules via focusing. *Annals of Pure and Applied Logic*, 173(5):103091, May 2022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168007222000057, doi:10.1016/j.apal.2022.103091.

38 Conor McBride. *Dependently Typed Functional Programs and their Proofs*. PhD thesis, University of Edinburgh, July 2000. URL: https://era.ed.ac.uk/handle/1842/374.

39 Paul-André Melliès and Noam Zeilberger. A bifibrational reconstruction of Lawvere's presheaf hyperdoctrine. *arXiv:1601.06098 [cs, math]*, August 2016. arXiv: 1601.06098. URL: http://arxiv.org/abs/1601.06098.

40 Robin Milner. The use of machines to assist in rigorous proof. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 312(1522):411–422, 1984. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1984.0067, arXiv:https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1984.0067, doi:10.1098/rsta.1984.0067.

41 Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, pages 625–635, Cham, 2021. Springer International Publishing.

42 Arnold Oostra. *Los gráficos Alfa de Peirce aplicados a la lógica intuicionista*. Number 2 in Cuadernos de Sistemática Peirceana. Centro de Sistemática Peirceana, 2010.

43 Arnold Oostra. *Gráficos existenciales Beta intuicionistas*. Number 3 in Cuadernos de Sistemática Peirceana. Centro de Sistemática Peirceana, 2011.

44 Arnold Oostra. Equivalence proof for intuitionistic existential alpha graphs. In *Diagrammatic Representation and Inference: 12th International Conference, Diagrams 2021, Virtual, September 28–30, 2021, Proceedings*, page 188–195, Berlin, Heidelberg, 2021. Springer-Verlag. doi:10.1007/978-3-030-86062-2_16.

45 Arnold Oostra. *Advances in Peircean Mathematics: The Colombian School*, chapter Intuitionistic and Geometrical Extensions of Peirce's Existential Graphs, pages 105–180. De Gruyter, 2022.

46 Charles Sanders Peirce. Prolegomena to an Apology for Pragmaticism. *The Monist*, 16(4):492–546, 1906. Publisher: Oxford University Press. URL: https://www.jstor.org/stable/27899680.

47 Brian Ritchie. *The Design and Implementation of an Interactive Proof Editor*. PhD thesis, 1988. Accepted: 2013-04-02T15:18:04Z Publisher: The University of Edinburgh. URL: https://era.ed.ac.uk/handle/1842/6607.

48 Don D. Roberts. *The Existential Graphs of Charles S. Peirce*. De Gruyter Mouton, Berlin, Boston, 1973. URL: https://doi.org/10.1515/9783110226225 [cited 2023-03-06], doi:doi:10.1515/9783110226225.

49 Benoit Rognier and Guillaume Duhamel. Présentation de la plateforme edukera. In *Vingt-septièmes Journées Francophones des Langages Applicatifs (JFLA 2016)*, 2016.

50 Sun-Joo Shin. *The Iconic Logic of Peirce's Graphs*. The MIT Press, 05 2002. doi:10.7551/mitpress/3633.001.0001.

51 Thoralf Skolem. Logisch-kombinatorische untersuchungen Über die erfüllbarkeit oder bewiesbarkeit mathematischer sätze nebst einem theorem Über dichte mengen. In Thoralf Skolem, editor, *Selected Works in Logic*. Universitetsforlaget, 1920.

52 John Sowa. Peirce's Tutorial on Existential Graphs. *Semiotica*, 186:345–394, 2011. doi:10.1515/semi.2011.060.

53 M. H. Stone. Topological representations of distributive lattices and Brouwerian logics. In *Časopis pro pěstování matematiky a fysiky*, volume 067, pages 1–25, 1938. ISSN: 1802-114X Issue: 1 Journal Abbreviation: Časopis Pěst. Mat. Fys. URL: https://dml.cz/handle/10338.dmlcz/124080, doi:10.21136/CPMF.1938.124080.

54 The Coq Development Team. The Coq Proof Assistant, 2022. doi:10.5281/zenodo.7313584.

55 Alwen Tiu. A Local System for Intuitionistic Logic. In Miki Hermann and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, Lecture Notes in Computer Science, pages 242–256, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11916277_17.

56 Fernando Zalamea. Lógica topológica. una introducción a los gráficos existenciales de peirce. Memorias del XIV Coloquio Distrital de Matemáticas y Estadística, 1997.

57 Fernando Zalamea. Pragmaticismo, gráficos y continuidad. hacia el lugar de c. s. peirce en la historia de la lógica. Mathesis 13, pp. 147–156, 1997.

58 Fernando Zalamea. Peirce's logic of continuity: Existential graphs and non-Cantorian continuum. *Review of Modern Logic*, 9(1-2):115–162, January 2003. Publisher: The Review of Modern Logic. URL: https://projecteuclid.org/journals/review-of-modern-logic/volume-9/issue-1-2/Peirces-logic-of-continuity--Existential-graphs-and-non-Cantorian/rml/1081173838.full.

59 J.J. Zeman. *The Graphical Logic of C. S. Peirce*. PhD thesis, University of Chicago, 1964. URL: https://books.google.fr/books?id=EOAqAQAAMAAJ.

60 Bohua Zhan, Zhenyan Ji, Wenfan Zhou, Chaozhu Xiang, Jie Hou, and Wenhui Sun. Design of point-and-click user interfaces for proof assistants. In *Formal Methods and Software Engineering: 21st International Conference on Formal Engineering Methods, ICFEM 2019, Shenzhen, China, November 5–9, 2019, Proceedings*, page 86–103, Berlin, Heidelberg, 2019. Springer-Verlag. doi:10.1007/978-3-030-32409-4_6.

A Comparison with EGs

In this section, we give a detailed comparison of the rules of the flower calculus with the illative transformations of Peirce's system Beta. To ease the presentation, we introduce an inductive syntax for the graphs of Beta based on our notion of garden.

▶ **Definition A.1** (β-graph). *The sets of β-nodes $\mathbf{N}_\beta$, β-gardens $\mathbf{\Gamma}_\beta$ and β-graphs $\mathbf{G}_\beta$ are defined mutually inductively:*

Atom *If $p \in \mathcal{P}$ and $\vec{\mathbf{x}} \in \mathcal{V}^{\mathsf{ar}(p)}$, then $p(\vec{\mathbf{x}}) \in \mathbf{N}_\beta$;*
Graph *If $G \subset \mathbf{N}_\beta$ is a finite multiset, then $G \in \mathbf{G}_\beta$.*
Garden *If $\mathbf{x} \subset \mathcal{V}$ is a finite set and $G \subset \mathbf{N}_\beta$ a finite multiset, then $\mathbf{x} \cdot G \in \mathbf{\Gamma}_\beta$;*
Cut *If $\gamma \in \mathbf{\Gamma}_\beta$, then $[\gamma] \in \mathbf{N}_\beta$.*

▶ Remark A.2. Note that a β-graph is defined as a multiset of β-nodes, just like a bouquet is a multiset of flowers. Then like in the flower calculus, and unlike in Peirce's original system, binders (LoIs) cannot appear at the top-level of SA.

▶ **Example A.3.** The lower graph in Figure 4a can be written in textual notation as the expression $[x \cdot R(x), [\cdot P(x)]]$.

We also adapt the notion of context to the Beta setting:

▶ **Definition A.4** (β-context). *β-contexts $\hat{G}$ are defined inductively by the following grammar:*

$$\hat{G}, \hat{H}, \hat{K} ::= G, \hat{g} \qquad\qquad \hat{g}, \hat{h}, \hat{k} ::= \square \mid [\hat{G}]$$

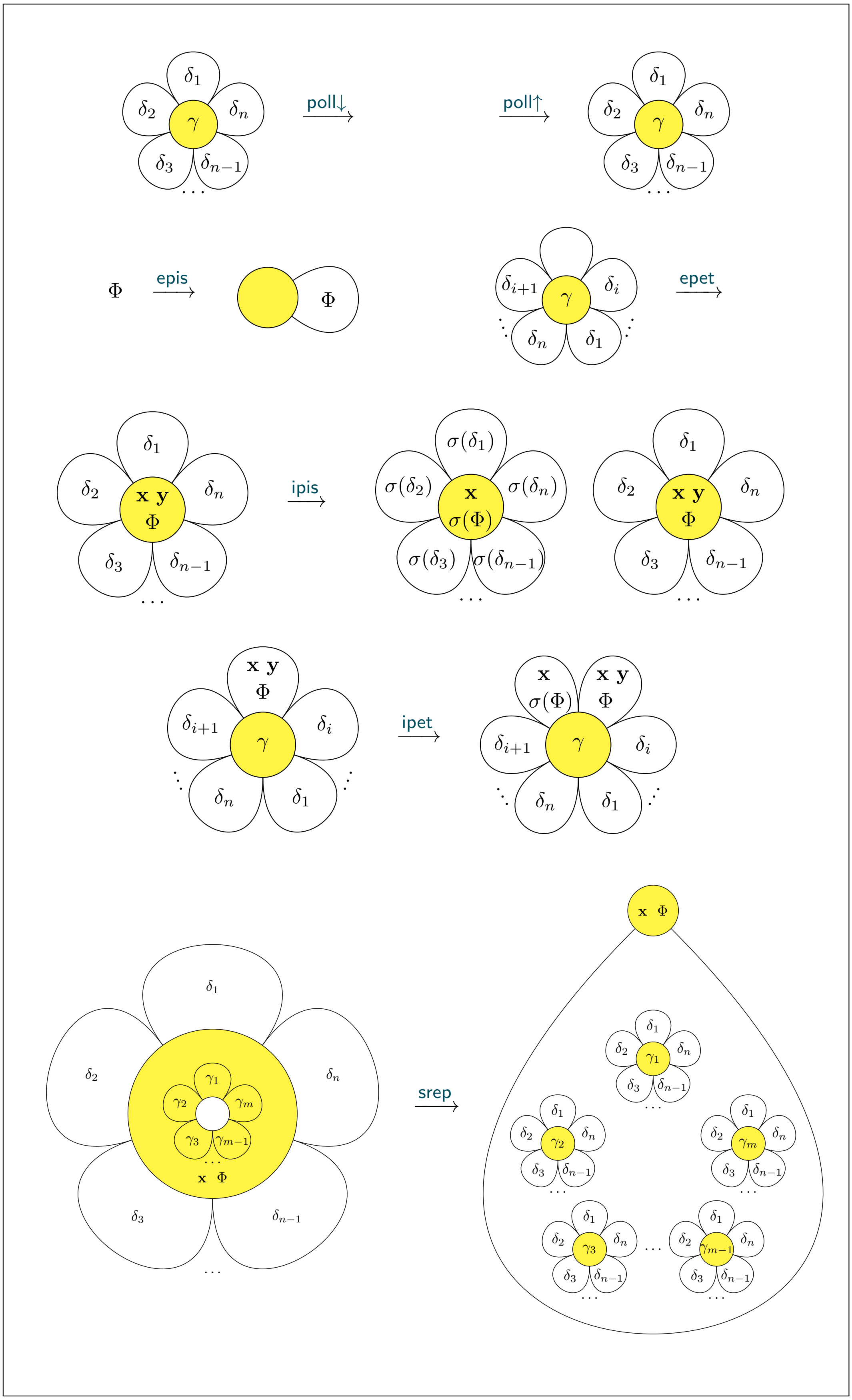


Figure 8 Graphical presentation of natural rules ❀

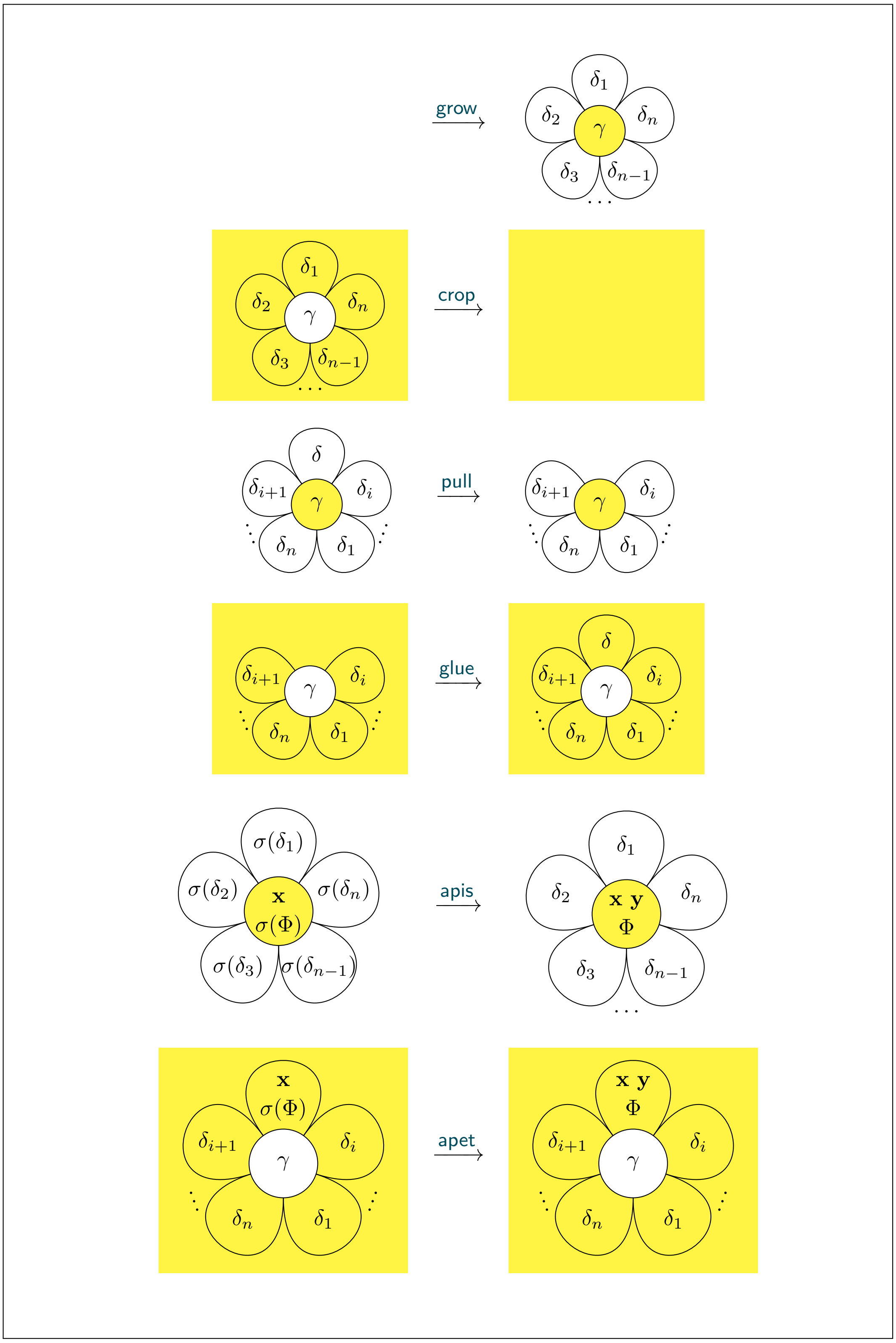
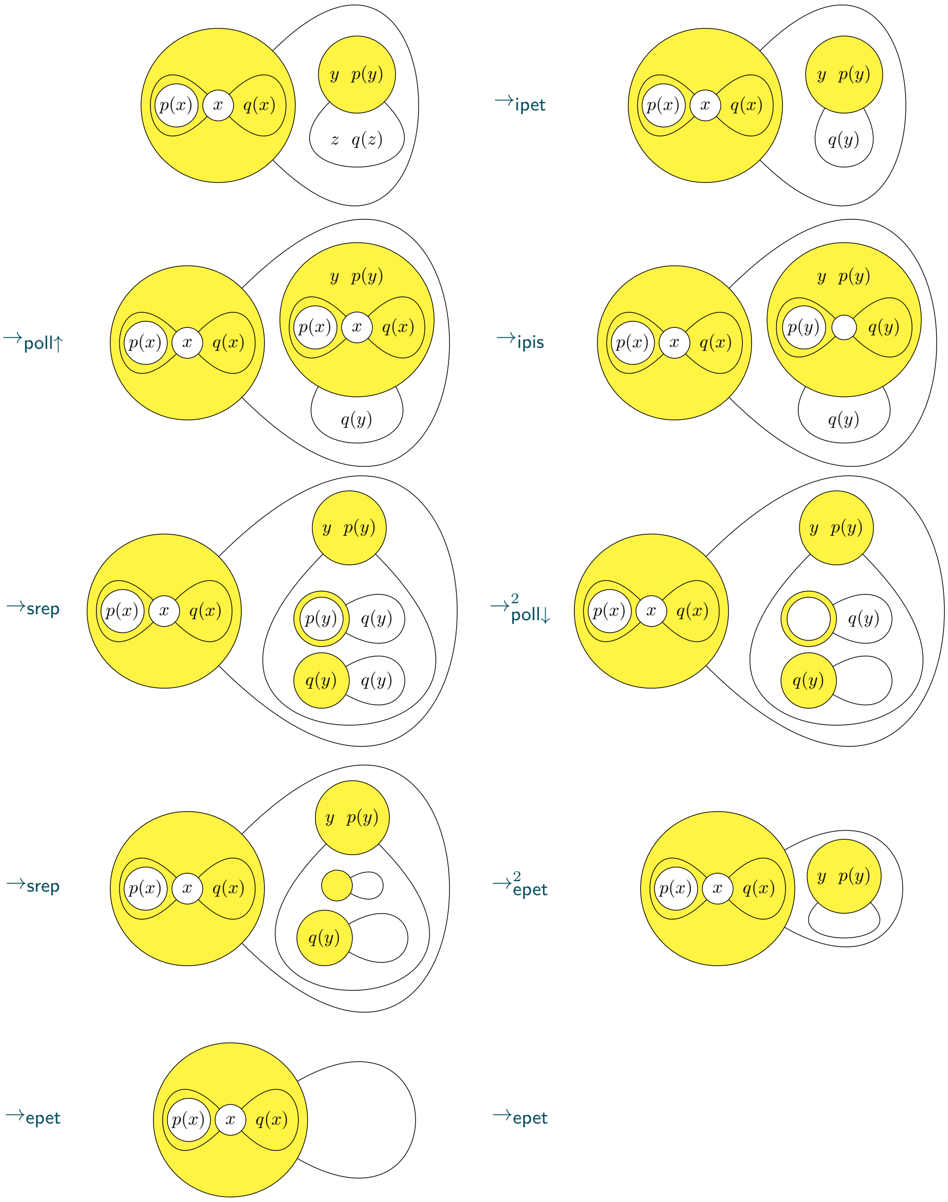


Figure 9 Graphical presentation of cultural rules ✂

$$
\begin{array}{ll}
& (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y)\ \mathsf{D}\ z \cdot q(z)) \\
\rightarrow_{\mathsf{ipet}} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y)\ \mathsf{D}\ q(y)) \\
\rightarrow_{\mathsf{poll}\uparrow} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y), (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ q(y)) \\
\rightarrow_{\mathsf{ipis}} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y), (\mathsf{D}\ (p(y)\ \mathsf{D}\ ); q(y))\ \mathsf{D}\ q(y)) \\
\rightarrow_{\mathsf{srep}} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y)\ \mathsf{D}\ ((p(y)\ \mathsf{D}\ )\ \mathsf{D}\ q(y)), (q(y)\ \mathsf{D}\ q(y))) \\
\rightarrow^{2}_{\mathsf{poll}\downarrow} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y)\ \mathsf{D}\ ((\mathsf{D}\ )\ \mathsf{D}\ q(y)), (q(y)\ \mathsf{D}\ \cdot)) \\
\rightarrow_{\mathsf{srep}} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y)\ \mathsf{D}\ (\mathsf{D}\ \cdot), (q(y)\ \mathsf{D}\ \cdot)) \\
\rightarrow^{2}_{\mathsf{epet}} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ (y \cdot p(y)\ \mathsf{D}\ \cdot) \\
\rightarrow_{\mathsf{epet}} & (x \cdot\ \mathsf{D}\ (p(x)\ \mathsf{D}\ ); q(x))\ \mathsf{D}\ \cdot \\
\rightarrow_{\mathsf{epet}} &
\end{array}
$$

(a) Textual presentation



(b) Graphical presentation

Figure 10 A natural proof in the flower calculus

$$\frac{\hat{K}\{G, \hat{H}\{\}\}}{\hat{K}\{G, \hat{H}\{G\}\}}\ \mathsf{Iter} \qquad \frac{\hat{K}\{G, \hat{H}\{G\}\}}{\hat{K}\{G, \hat{H}\{\}\}}\ \mathsf{Deit}$$

$$\frac{\hat{K}^-\{\}}{\hat{K}^-\{G\}}\ \mathsf{Ins} \qquad \frac{\hat{K}^+\{G\}}{\hat{K}^+\{\}}\ \mathsf{Del}$$

$$\frac{\hat{K}\{G\}}{\hat{K}\{[\cdot [\cdot G]]\}}\ \mathsf{Dcut}{\downarrow} \qquad \frac{\hat{K}\{[\cdot [\cdot G]]\}}{\hat{K}\{G\}}\ \mathsf{Dcut}{\uparrow}$$

$$\frac{\hat{K}^+\{[\mathbf{x}, y \cdot G]\}}{\hat{K}^+\{[\mathbf{x} \cdot G\{z/y\}]\}}\ \mathsf{Unif}{\downarrow} \qquad \frac{\hat{K}^-\{[\mathbf{x} \cdot G\{z/y\}]\}}{\hat{K}^-\{[\mathbf{x}, y \cdot G]\}}\ \mathsf{Unif}{\uparrow}$$

(a) Rules of Beta

$$\frac{\Phi}{\cdot \ \mathsf{D} \ \cdot \Phi}\ \mathsf{epis}{\downarrow}$$

(b) Converse of epis rule

Figure 11 Comparing rules

▶ **Definition A.5** (Polarity). *The number of* inversions $\mathsf{inv}(\hat{G})$ *of a β-context* $\hat{G}$ *is:*

$$\mathsf{inv}(\square) = 0 \qquad \mathsf{inv}(G, \hat{g}) = \mathsf{inv}(\hat{g}) \qquad \mathsf{inv}([\hat{G}]) = 1 + \mathsf{inv}(\hat{G})$$

The definitions of free variables, bound variables and substitutions can also be adapted straightforwardly.

The set of rules of Beta is given in Figure 11a. Iter, Deit, Ins and Del correspond respectively to the Iteration, Deiteration, Insertion and Deletion principles of Alpha, while Dcut↓ and Dcut↑ capture the Double-cut principle. As for Unif↓ and Unif↑, they correspond roughly to the principles of Insertion and Deletion applied to LoIs. Indeed in their bottom-up reading, Unif↑ and Unif↓ can be understood as capturing respectively the operations of *unification* and *anti-unification* on two variables:

Unification substituting z for y and removing the binder for y is equivalent in purpose to adding a new LoI between the outermost point of the LoI associated to y, and some point of the LoI associated to z in the same area (which is assumed to exist by well-scopedness);

Anti-unification substituting y for z and adding a binder for y is equivalent in purpose to severing the LoI associated to z at the location where the binder for y is introduced.

There is no need to formulate an equivalent of the Iteration and Deiteration principles for LoIs. Indeed, their purpose is to manage *locally* the *extension* of a line. With binders, the notion of extension is replaced with that of *scope*, which is handled *globally* and automatically in the definition of substitutions.

Let us now review the rules of the flower calculus in more detail, starting with the fragment that is a direct adaptation of the rules of Beta:

Blank Antecedent (epis) It allows to enclose any bouquet in a petal attached to an (e)mpty (pis)til. This is a weaker, intuitionistic version of Dcut↑, that was already identified by Peirce as the "collapsing of a scroll's walls' [46, p. 534], and is called the rule of Blank

Antecedant in [36]. The converse rule that would correspond to Dcut↓ — rule epis↓ in Figure 11b — is actually shown to be *admissible* by our completeness theorem[11].

(De)iteration (poll↓, poll↑) The (poll)ination rules poll↓ and poll↑ correspond respectively to Iter and Deit, but reformulated with the pollination relation (Definition 6.3). In fact in their textual presentation (Figure 7), they are more general than (de)iteration rules, because $\Phi \succ \hat{\Xi}$ allows the pollinating bouquet Φ to be *scattered* in the context $\hat{\Xi}$, i.e. its flowers need not be located in the same area. On the contrary in their graphical presentation (Figure 8), they are less general since only one flower can be pollinated at a time, rather than an entire bouquet of flowers residing in the same area. But it is easy to see that all these variants are equivalent in deductive power, since the pollination of a bouquet (however scattered) can always be simulated by the successive pollinations of each of its flowers.

Insertion/Deletion (grow, crop, pull, glue) They correspond to Ins and Del, but have doubled in number to account for the syntactic distinction between pistils and petals. More precisely, rules grow and crop allow to insert and delete entire flowers, while rules pull and glue deal with petals. As for pollination rules, manipulating single flowers/petals (graphical version) or entire bouquets/corollas (textual version) does not change the deductive power of the rules.

Unification (ipis, ipet, apis, apet) Rules ipis and ipet allow to (i)nstantiate a sprinkler located respectively in a (pis)til ($\forall$) and a (pet)al ($\exists$) with an arbitrary substitution, while rules apis and apet do the opposite operation of (a)bstracting a set of variables by introducing a sprinkler. They correspond respectively to a generalization of Unif↑ and Unif↓, where the variable substitution $\{z/y\}$ becomes an arbitrary substitution σ. Once again, we have twice the amount of rules to account for the pistil/petal distinction, which is not surprising since in the LoI syntax of EGs, they are special cases of Insertion/Deletion. Note that for the instantiation rules ipis/ipet to be invertible, we duplicate the whole flower/petal where the sprinkler occurs, mirroring what is done in multi-conclusion sequent calculi.

The last two rules mainly handle the behavior of disjunctive and absurd statements, i.e. flowers with respectively $n \geq 2$ and $n = 0$ petals, and are closer to sequent-style introduction/elimination rules:

Disjunction Introduction (epet) It allows to erase any flower with an (e)mpty (pet)al. According to Oostra [45, p. 109], Peirce already identified epet as a component of his decision procedure for Alpha (it is simply called “Operation 1” in [45]). This is no coincidence, since we precisely came up with this rule when trying to design a decision procedure for flowers.

Disjunction/Falsehood Elimination (srep) It corresponds to a n-ary generalization of the left introduction rule for disjunction in sequent calculus, the 0-ary case capturing falsehood elimination (*ex falso quodlibet*) as illustrated in the proof of Figure 10. The binary case is also used in the intuitionistic EGs system of [36] together with its converse, which is also shown to be admissible by our completeness theorem. The name srep is short for (s)elf-(rep)roduction, which is more clearly visualized in the graphical version of the rule in Figure 8.

[11] Interestingly, although an equivalent of epis↓ is included in [36], there is no mention of it by Peirce in [46]. It may be a sign that Peirce already intuited its admissibility, or at least considered this direction of the transformation unworthy of attention.

B Soundness

In this section, we show that every rule of the flower calculus is *sound* with respect to our Kripke semantics for flowers, and thus that $\vdash \phi$ implies $\vDash \phi$ for every ϕ. We start with a few trivial facts about Definition 5.5:

▶ **Observation B.1** (Associativity). $f \,|_R\, g \,|_S\, h = f \,|_{R \cup S}\, (g \,|_S\, h)$.

▶ **Observation B.2** (Commutativity). *If* $R \cap S = \varnothing$ *then* $f \,|_R\, g \,|_S\, h = f \,|_S\, h \,|_R\, g$.

▶ **Observation B.3** (Agreement). *If* $f(x) = g(x)$ *for all* $x \in R$ *then* $h \,|_R\, f = h \,|_R\, g$.

▶ **Observation B.4** (Idempotency). $f \,|_R\, f = f$.

Semantic entailment is obviously a reflexive and transitive relation:

▶ **Observation B.5** (Reflexivity). $\Phi \vDash \Phi$.

▶ **Observation B.6** (Transitivity). *If* $\Phi \vDash \Psi$ *and* $\Psi \vDash \Xi$, *then* $\Phi \vDash \Xi$.

The two following lemmas will be useful to reason on the forcing relation (Definition 7.5):

▶ **Lemma B.7** (Monotonicity). *If* $w \leq w'$ *and* $w \Vdash \phi\,[e]$ *then* $w' \Vdash \phi\,[e]$.

Proof. By a straightforward induction on $|\phi|$. ◀

▶ **Lemma B.8** (Mirroring). $w \Vdash \sigma(\phi)\,[e]$ *iff* $w \Vdash \phi\,[e \,|_{\mathbf{x}}\, [\![\sigma]\!]_e]$ *for* $\sigma : \mathbf{x}$ *capture-avoiding in* ϕ *and* $\mathbf{x} \cap \mathsf{bv}(\phi) = \varnothing$.

Proof. By induction on $|\phi|$.

Base case Suppose $\phi = p(\vec{\mathbf{y}})$. We show that $[\![\sigma(\vec{\mathbf{y}})]\!]_e \in [\![p]\!]_w$ iff $[\![\vec{\mathbf{y}}]\!]_{e|_{\mathbf{x}}[\![\sigma]\!]_e} \in [\![p]\!]_w$ by proving that $[\![x]\!]_{e|_{\mathbf{x}}[\![\sigma]\!]_e} = [\![\sigma(x)]\!]_e$ for any variable x. Either:

- $x \in \mathbf{x}$, and $[\![x]\!]_{e|_{\mathbf{x}}[\![\sigma]\!]_e} = [\![\sigma]\!]_e(x) = [\![\sigma(x)]\!]_e$; or
- $x \notin \mathbf{x}$, and $[\![x]\!]_{e|_{\mathbf{x}}[\![\sigma]\!]_e} = e(x) = [\![x]\!]_e = [\![\sigma(x)]\!]_e$.

Recursive case Suppose $\phi = \mathbf{y} \cdot \Phi \supset \{\mathbf{z}_i \cdot \Psi_i\}_i^n$. We show that $w \Vdash \mathbf{y} \cdot \sigma(\Phi) \supset \{\mathbf{z}_i \cdot \sigma(\Psi_i)\}_i^n\,[e]$ implies $w \Vdash \mathbf{y} \cdot \Phi \supset \{\mathbf{z}_i \cdot \Psi_i\}_i^n\,[e \,|_{\mathbf{x}}\, [\![\sigma]\!]_e]$, the argument working in both directions. Let $w' \geq w$ and e' a w'-evaluation such that $w' \Vdash \Phi\,[e \,|_{\mathbf{x}}\, [\![\sigma]\!]_e \,|_{\mathbf{y}}\, e']$. Since σ is capture-avoiding in ϕ, we know that $\mathsf{fv}(\sigma(x)) \cap \mathbf{y} = \varnothing$, and thus $[\![\sigma]\!]_e(x) = [\![\sigma(x)]\!]_e = [\![\sigma(x)]\!]_{e|_{\mathbf{y}}e'} = [\![\sigma]\!]_{e|_{\mathbf{y}}e'}(x)$ for any $x \in \mathbf{x}$. Hence by Observation B.3 $w' \Vdash \Phi\,[e \,|_{\mathbf{x}}\, [\![\sigma]\!]_{e|_{\mathbf{y}}e'} \,|_{\mathbf{y}}\, e']$, and since by hypothesis $\mathbf{x} \cap \mathbf{y} = \varnothing$ we obtain $w' \Vdash \Phi\,[e \,|_{\mathbf{y}}\, e' \,|_{\mathbf{x}}\, [\![\sigma]\!]_{e|_{\mathbf{y}}e'}]$ by Observation B.2. Then by IH we get $w' \Vdash \sigma(\Phi)\,[e \,|_{\mathbf{y}}\, e']$, and thus by hypothesis $w' \Vdash \sigma(\Psi_i)\,[e \,|_{\mathbf{y}}\, e' \,|_{\mathbf{z}_i}\, e'']$ for some $1 \leq i \leq n$ and w'-evaluation e''. Again by IH we get $w' \Vdash \Psi_i\,[e \,|_{\mathbf{y}}\, e' \,|_{\mathbf{z}_i}\, e'' \,|_{\mathbf{x}}\, [\![\sigma]\!]_{e|_{\mathbf{y}}e'|_{\mathbf{z}_i}e''}]$, and since σ is capture-avoiding in ϕ we have $\mathsf{fv}(\sigma(x)) \cap \mathbf{z}_i = \varnothing$ for any $x \in \mathbf{x}$, and thus $w' \Vdash \Psi_i\,[e \,|_{\mathbf{y}}\, e' \,|_{\mathbf{z}_i}\, e'' \,|_{\mathbf{x}}\, [\![\sigma]\!]_e]$ by Observation B.3. Finally by hypothesis $\mathbf{x} \cap \mathbf{z}_i = \varnothing$, thus we can conclude that $w' \Vdash \Psi_i\,[e \,|_{\mathbf{x}}\, [\![\sigma]\!]_e \,|_{\mathbf{y}}\, e' \,|_{\mathbf{z}_i}\, e'']$ by Observation B.2.

◀

The following *functoriality* lemma is at the heart of every deep inference formalism. It requires an induction principle for *contexts*:

▶ **Definition B.9** (Depth). *The* depth $|\hat{\Phi}|$ *of a context* $\hat{\Phi}$ *is defined recursively by:*

$$|\square| = 0 \qquad |\Psi, \hat{\phi}| = |\hat{\phi}| \qquad |\mathbf{x} \cdot \hat{\Phi} \supset \Delta| = |\gamma \supset \mathbf{x} \cdot \hat{\Phi}; \Delta| = 1 + |\hat{\Phi}|$$

▶ **Lemma B.10** (Functoriality). *If $\Phi \vDash \Psi$, then for any $\hat{\Xi}$ either $\hat{\Xi}\{\Phi\} \vDash \hat{\Xi}\{\Psi\}$ if $\hat{\Xi}$ is positive, or $\hat{\Xi}\{\Psi\} \vDash \hat{\Xi}\{\Phi\}$ if $\hat{\Xi}$ is negative.*

Proof. By induction on $|\hat{\Xi}|$. ◀

▶ **Lemma B.11** (Weakening). $\Phi \vDash \varnothing$.

Proof. Trivial by Definition 7.5. ◀

▶ **Lemma B.12** (Co-weakening). $\gamma \supset \Delta \vDash \gamma \supset \Gamma ; \Delta$.

Proof. Let $\gamma = \mathbf{x} \cdot \Phi$, w a world in some Kripke structure $\mathcal{K}$, $w' \geq w$, e a w-evaluation and e' a w'-evaluation such that $w \Vdash \gamma \supset \Delta\,[e]$ and $w' \Vdash \Phi\,[e \,|_{\mathbf{x}}\, e']$. Then by hypothesis there must exist some $\mathbf{y} \cdot \Psi \in \Delta$ and w'-evaluation e'' such that $w' \Vdash \Psi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$, and thus we can conclude. ◀

The less obvious rules in terms of soundness are the *pollination* rules $\{\mathsf{poll}{\downarrow}, \mathsf{poll}{\uparrow}\}$, because of the arbitrary context $\hat{\Xi}$ and reliance on the pollination relation.

▶ **Lemma B.13** (Cross-pollination). $\Phi, \hat{\Xi}\{\Phi\} \dashv\vDash \Phi, \hat{\Xi}\{\}$.

Proof. Let w a world in some Kripke structure $\mathcal{K}$, and e a w-evaluation. We show that $w \Vdash \Phi, \hat{\Xi}\{\Phi\}\,[e]$ iff $w \Vdash \Phi, \hat{\Xi}\{\}\,[e]$ by induction on $|\hat{\Xi}|$.

Base case Suppose $\hat{\Xi} = \Xi', \Box$. Then we trivially have $w \Vdash \Phi, \Xi', \Phi\,[e]$ iff $w \Vdash \Phi, \Xi'\,[e]$ by Definition 7.5.

Recursive case We distinguish two cases:

Pistil Suppose $\hat{\Xi} = \Xi', (\mathbf{x} \cdot \hat{\Xi}_0 \supset \Delta)$.

1. Suppose $w \Vdash \Phi, \hat{\Xi}\{\Phi\}\,[e]$. Then $w \Vdash \Phi\,[e]$, $w \Vdash \Xi'\,[e]$ and $w \Vdash \mathbf{x} \cdot \hat{\Xi}_0\{\Phi\} \supset \Delta\,[e]$. Thus it remains to show that $w \Vdash \mathbf{x} \cdot \hat{\Xi}_0\{\} \supset \Delta\,[e]$. Let $w' \geq w$ and e' a w'-evaluation such that $w' \Vdash \hat{\Xi}_0\{\}\,[e \,|_{\mathbf{x}}\, e']$. By IH we have $\Phi, \hat{\Xi}_0\{\} \vDash \Phi, \hat{\Xi}_0\{\Phi\}$, and thus by Lemma B.10 $\mathbf{x} \cdot \Phi, \hat{\Xi}_0\{\Phi\} \supset \Delta \vDash \mathbf{x} \cdot \Phi, \hat{\Xi}_0\{\} \supset \Delta$. By Lemma B.11 and Lemma B.10 we have $w \Vdash \mathbf{x} \cdot \Phi, \hat{\Xi}_0\{\Phi\} \supset \Delta\,[e]$, and thus $w \Vdash \mathbf{x} \cdot \Phi, \hat{\Xi}_0\{\} \supset \Delta\,[e]$. Then since $w' \Vdash \hat{\Xi}_0\{\}\,[e \,|_{\mathbf{x}}\, e']$, and since by Lemma B.7 (and the fact that $\mathbf{x} \cap \mathsf{fv}(\Phi) = \varnothing$) we have $w' \Vdash \Phi\,[e \,|_{\mathbf{x}}\, e']$, we can conclude that there are some $\mathbf{y} \cdot \Psi \in \Delta$ and w'-evaluation e'' such that $w' \Vdash \Psi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$.
2. $\Phi, \hat{\Xi}\{\} \vDash \Phi, \hat{\Xi}\{\Phi\}$ holds by the same argument in the other direction.

Petal Suppose $\hat{\Xi} = \Xi', (\mathbf{x} \cdot \Psi \supset \mathbf{y} \cdot \hat{\Xi}_0 ; \Delta)$.

1. Suppose $x \Vdash \Phi, \hat{\Xi}\{\Phi\}\,[e]$. Then $w \Vdash \Phi\,[e]$, $w \Vdash \Xi'\,[e]$ and $w \Vdash \mathbf{x} \cdot \Psi \supset \mathbf{y} \cdot \hat{\Xi}_0\{\Phi\} ; \Delta\,[e]$. Thus it remains to show that $w \Vdash \mathbf{x} \cdot \Psi \supset \mathbf{y} \cdot \hat{\Xi}_0\{\} ; \Delta\,[e]$. Let $w' \geq w$ and e' a w'-evaluation such that $w' \Vdash \Psi\,[e \,|_{\mathbf{x}}\, e']$. Then we can deduce that there exists a w'-evaluation e'' such that either:
   - $w' \Vdash \Psi'\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}'}\, e'']$ for some $\mathbf{y}' \cdot \Psi' \in \Delta$, and we conclude immediately;
   - or $w' \Vdash \hat{\Xi}_0\{\Phi\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$. By Lemma B.7 (and the fact that $\mathbf{x} \cap \mathsf{fv}(\Phi) = \varnothing$ and $\mathbf{y} \cap \mathsf{fv}(\Phi) = \varnothing$) we have $w' \Vdash \Phi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$, and thus $w' \Vdash \Phi, \hat{\Xi}_0\{\Phi\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$. Then by IH we have $w' \Vdash \Phi, \hat{\Xi}_0\{\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$, and thus we can conclude in particular that $w' \Vdash \hat{\Xi}_0\{\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$.
2. $\Phi, \hat{\Xi}\{\} \vDash \Phi, \hat{\Xi}\{\Phi\}$ holds by the same argument in the other direction.

◀

▶ **Lemma B.14** (Pollination). *If $\Phi \succ \hat{\Xi}$, then $\hat{\Xi}\{\Phi\} \dashv\vDash \hat{\Xi}\{\}$.*

Proof. We show that $\phi \succ \hat{\Xi}$ implies $\hat{\Xi}\{\phi\} \Dashv\vDash \hat{\Xi}\{\}$ for any flower ϕ and context $\hat{\Xi}$: then assuming that $\Phi = \phi_1, \ldots, \phi_n$, we get

$$\underbrace{\hat{\Xi}\{\phi_1, \ldots, \phi_n\} \Dashv\vDash \hat{\Xi}\{\phi_2, \ldots, \phi_n\} \Dashv\vDash \ldots \Dashv\vDash \hat{\Xi}\{\}}_{n \text{ times}}$$

and conclude by Observation B.6.

By Definition 6.3, there are a bouquet Ψ and two contexts $\hat{\Xi}', \hat{\Xi}_0$ such that one of the two following cases holds:

Cross-pollination $\hat{\Xi} = \hat{\Xi}'\{\Psi, \phi, \hat{\Xi}_0\}$. Then $\phi, \hat{\Xi}_0\{\phi\} \Dashv\vDash \phi, \hat{\Xi}_0\{\}$ by Lemma B.13, and we conclude by Lemma B.10.

Self-pollination $\hat{\Xi} = \hat{\Xi}'\{\mathbf{x} \cdot \Psi, \phi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0 ; \Delta\}$ for some $\mathbf{x}, \mathbf{y}, \Delta$. Let w a world in some Kripke structure $\mathcal{K}$ and e a w-evaluation. We show that $w \Vdash \mathbf{x} \cdot \Psi, \phi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0\{\phi\} ; \Delta\,[e]$ iff $w \Vdash \mathbf{x} \cdot \Psi, \phi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0\{\} ; \Delta\,[e]$, and conclude by Lemma B.10.

1. Suppose that $w \Vdash \mathbf{x} \cdot \Psi, \phi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0\{\phi\} ; \Delta\,[e]$, and let $w' \geq w$ and e' a w'-evaluation such that $w' \Vdash \Psi, \phi\,[e \,|_{\mathbf{x}}\, e']$. Then we can deduce that there exists a w'-evaluation e'' such that either:
   - $w' \Vdash \Psi'\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}'}\, e'']$ for some $\mathbf{y}' \cdot \Psi' \in \Delta$, and we conclude immediately;
   - or $w' \Vdash \hat{\Xi}_0\{\phi\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$. Since $\mathsf{fv}(\phi) \cap \mathbf{y} = \varnothing$ we have $w' \Vdash \phi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$, and thus $w' \Vdash \phi, \hat{\Xi}_0\{\phi\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$. Then by Lemma B.13 we have $w' \Vdash \phi, \hat{\Xi}_0\{\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$, and thus we can conclude in particular that $w' \Vdash \hat{\Xi}_0\{\}\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, e'']$.
2. $\mathbf{x} \cdot \Psi, \phi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0\{\} ; \Delta \vDash \mathbf{x} \cdot \Psi, \phi \mathbin{\mathsf{D}} \mathbf{y} \cdot \hat{\Xi}_0\{\phi\} ; \Delta$ holds by the same argument in the other direction.

◀

Proving the soundness of rules involving binders (ipis, ipet, apis, apet) is also quite tedious, which can be understood as stemming from the fact that substitutions simulate the complex dynamics of the LoIs of EGs in a *global* rather than local way. In particular, one needs to be careful about the scope of bound variables, which in EGs would be handled locally with (de)iteration rules on LoIs.

▶ **Lemma B.15** (Universal instantiation). *If $\sigma : \mathbf{y}$ is capture-avoiding in $\Phi \mathbin{\mathsf{D}} \Delta$, then* $\mathbf{x}, \mathbf{y} \cdot \Phi \mathbin{\mathsf{D}} \Delta \vDash \mathbf{x} \cdot \sigma(\Phi) \mathbin{\mathsf{D}} \sigma(\Delta)$.

Proof. Let w a world in some Kripke structure $\mathcal{K}$, $w' \geq w$, e a w-evaluation and e' a w'-evaluation such that $w \Vdash \mathbf{x}, \mathbf{y} \cdot \Phi \mathbin{\mathsf{D}} \Delta\,[e]$ and $w' \Vdash \sigma(\Phi)\,[e \,|_{\mathbf{x}}\, e']$. Therefore $w' \Vdash \Phi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{x}}e'}]$ by Lemma B.8, and thus $w' \Vdash \Phi\,[e \,|_{\mathbf{x}\cup\mathbf{y}}\, (e' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{x}}e'})]$ by Observation B.1. Then by hypothesis, there must be some $\mathbf{z} \cdot \Psi \in \Delta$ and w'-evaluation e'' such that $w' \Vdash \Psi\,[e \,|_{\mathbf{x}\cup\mathbf{y}}\, (e' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{x}}e'}) \,|_{\mathbf{z}}\, e'']$, and thus $w' \Vdash \Psi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{x}}e'} \,|_{\mathbf{z}}\, e'']$. Since σ is capture-avoiding in $\Phi \mathbin{\mathsf{D}} \Delta$, we know that for any $x \in \mathbf{y}$ we have $\mathsf{fv}(\sigma(x)) \cap \mathbf{z} = \varnothing$, and thus $[\![\sigma(x)]\!]_{e|_{\mathbf{x}}e'|_{\mathbf{z}}e''} = [\![\sigma(x)]\!]_{e|_{\mathbf{x}}e'}$. Hence by Observation B.3 and Observation B.2 we get $w' \Vdash \Psi\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{z}}\, e'' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{x}}e'|_{\mathbf{z}}e''}]$, and by Lemma B.8 we conclude that $w' \Vdash \sigma(\Psi)\,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{z}}\, e'']$. ◀

▶ **Lemma B.16** (Existential instantiation). *If $\sigma : \mathbf{y}$ is capture-avoiding in Φ, then $\gamma \mathbin{\mathsf{D}} \mathbf{x} \cdot \sigma(\Phi) ; \Delta \vDash \gamma \mathbin{\mathsf{D}} \mathbf{x}, \mathbf{y} \cdot \Phi ; \Delta$.*

Proof. Let $\gamma = \mathbf{z} \cdot \Xi$, and w a world in some Kripke structure $\mathcal{K}$, $w' \geq w$, e a w-evaluation and e' a w'-evaluation such that $w \Vdash \gamma \mathbin{\mathsf{D}} \mathbf{x} \cdot \sigma(\Phi) ; \Delta\,[e]$ and $w' \Vdash \Xi\,[e \,|_{\mathbf{z}}\, e']$. Then by hypothesis, there must be some w'-evaluation e'' such that either:

- $w' \Vdash \Xi' \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{z}'}\, e'']$ for some $\mathbf{z}' \cdot \Xi' \in \Delta$, and we conclude immediately;
- or $w' \Vdash \sigma(\Phi) \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{x}}\, e'']$. Then by Lemma B.8 we have $w' \Vdash \Phi \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{x}}\, e'' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{z}}e'|_{\mathbf{x}}e''}]$, and thus we can conclude with $w' \Vdash \Phi \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{x} \cup \mathbf{y}}\, (e'' \,|_{\mathbf{y}}\, [\![\sigma]\!]_{e|_{\mathbf{z}}e'|_{\mathbf{x}}e''})]$ by Observation B.3.

◀

We are now equipped with enough lemmas to prove the soundness of each rule, starting with the *shallow* version of natural rules. In fact we are able to prove more: that every ❀-rule is *invertible*, i.e. its conclusion entails its premiss.

▶ **Lemma B.17** (Shallow soundness). *If $\Phi \rightharpoonup_{❀} \Psi$, then $\Phi \dashv\Vdash \Psi$.*

Proof. Let w a world in some Kripke structure $\mathcal{K}$, $w' \geq w$, e a w-evaluation and e' a w'-evaluation. We proceed by inspection of every ❀-rule.

poll↓, poll↑ By Lemma B.14.

epis

1. Suppose that $w \Vdash \Phi \,[e]$. Then by Lemma B.7 we have $w' \Vdash \Phi \,[e]$, and thus we can conclude for instance with $w' \Vdash \Phi \,[e \,|_{\varnothing}\, e' \,|_{\varnothing}\, e]$.
2. Suppose that $w \Vdash \;\mathsf{D}\; \Phi \,[e]$. Then since we trivially have $w \geq w$ and $w \Vdash \varnothing \,[e \,|_{\varnothing}\, e]$, we get that $w \Vdash \Phi \,[e \,|_{\varnothing}\, e \,|_{\varnothing}\, e'']$ for some w-evaluation e'', and thus $w \Vdash \Phi \,[e]$.

epet Let $\gamma = \mathbf{x} \cdot \Phi$. We trivially have that $w' \Vdash \varnothing \,[e \,|_{\mathbf{x}}\, e' \,|_{\varnothing}\, e]$, and thus can conclude.

ipis We trivially have $\mathbf{x}, \mathbf{y} \cdot \Phi \;\mathsf{D}\; \Delta \vDash \mathbf{x}, \mathbf{y} \cdot \Phi \;\mathsf{D}\; \Delta$ by Observation B.5, and thus we can conclude by Lemma B.15.

ipet The first direction is trivial by Lemma B.12. In the other direction, let $\gamma = \mathbf{z} \cdot \Xi$, and suppose that $w \Vdash \gamma \;\mathsf{D}\; \mathbf{x} \cdot \sigma(\Phi) \,;\, \mathbf{x}, \mathbf{y} \cdot \Phi \,;\, \Delta \,[e]$ and $w' \Vdash \Xi \,[e \,|_{\mathbf{z}}\, e']$. Then there must be some w'-evaluation e'' such that either:

- $w' \Vdash \Xi' \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{z}'}\, e'']$ for some $\mathbf{z}' \cdot \Xi' \in \Delta$, and we conclude immediately;
- $w' \Vdash \Phi \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{x} \cup \mathbf{y}}\, e'']$, and we also conclude immediately;
- or $w' \Vdash \sigma(\Phi) \,[e \,|_{\mathbf{z}}\, e' \,|_{\mathbf{x}}\, e'']$, and we conclude with the same argument as in the proof of Lemma B.16.

srep Let $\gamma_i = \mathbf{y}_i \cdot \Psi_i$ for $1 \leq i \leq n$.

1. Suppose that $w \Vdash \mathbf{x} \cdot \Phi, (\mathsf{D}\; \{\gamma_i\}_i^n) \;\mathsf{D}\; \Delta \,[e]$ and $w' \Vdash \Phi \,[e \,|_{\mathbf{x}}\, e']$. We show that $w' \Vdash \gamma_i \;\mathsf{D}\; \Delta \,[e \,|_{\mathbf{x}}\, e']$ for all $1 \leq i \leq n$, i.e. for every $w'' \geq w'$ and w''-evaluation e'', $w'' \Vdash \Psi_i \,[e|_{\mathbf{x}}e'|_{\mathbf{y}_i}e'']$ implies that there is some $\mathbf{z}\cdot\Xi \in \Delta$ and w''-evaluation e''' such that $w'' \Vdash \Xi \,[e|_{\mathbf{x}}e'|_{\mathbf{y}_i}e''|_{\mathbf{z}}e''']$. By assumption, Lemma B.7 and the fact that $\mathsf{fv}(\Phi) \cap \mathbf{y}_i = \varnothing$, we have $w'' \Vdash \Phi \,[e \,|_{x}\, e' \,|_{\mathbf{y}_i}\, e'']$. Also since $w'' \Vdash \Psi_i \,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}_i}\, e'']$ we immediately get $w'' \Vdash \mathsf{D}\; \{\gamma_i\}_i^n \,[e \,|_{\mathbf{x}}\, e']$, and thus $w'' \Vdash \mathsf{D}\; \{\gamma_i\}_i^n \,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}_i}\, e'']$ since $\mathsf{fv}(\mathsf{D}\; \{\gamma_i\}_i^n) \cap \mathbf{y}_i = \varnothing$. Thus by Observation B.1 we have $w'' \Vdash \Phi, (\mathsf{D}\; \{\gamma_i\}_i^n) \,[e|_{\mathbf{x} \cup \mathbf{y}_i}(e'|_{\mathbf{y}_i}e'')]$, and by hypothesis (and the fact that $w'' \geq w$ by transitivity) we obtain that $w'' \Vdash \Xi \,[e|_{\mathbf{x} \cup \mathbf{y}_i}(e'|_{\mathbf{y}_i}e'')|_{\mathbf{z}}e''']$ for some $\mathbf{z} \cdot \Xi \in \Delta$ and w''-evaluation e'''. Then we conclude again by Observation B.1.
2. Suppose that $w \Vdash \mathbf{x} \cdot \Phi \;\mathsf{D}\; \{\gamma_i \;\mathsf{D}\; \Delta\}_i^n \,[e]$ and $w' \Vdash \Phi, (\mathsf{D}\; \{\gamma_i\}_i^n) \,[e \,|_{\mathbf{x}}\, e']$. Then there must be some $1 \leq i \leq n$ and w'-evaluation e'' such that $w' \Vdash \Psi_i \,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}_i}\, e'']$, and for all $1 \leq j \leq n$ we know that $w' \Vdash \gamma_j \;\mathsf{D}\; \Delta \,[e \,|_{\mathbf{x}}\, e']$. Thus since $w' \leq w'$ by reflexivity, there must be some $\mathbf{z} \cdot \Xi \in \Delta$ and w'-evaluation e''' such that $w' \Vdash \Xi \,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}_i}\, e'' \,|_{\mathbf{z}}\, e''']$, and we can conclude with $w' \Vdash \Xi \,[e \,|_{\mathbf{x}}\, e' \,|_{\mathbf{y}_i \cup \mathbf{z}}\, (e'' \,|_{\mathbf{z}}\, e''')]$ by Observation B.1.

◀

Then the soundness of the contextual closure of natural rules follows immediately from functoriality:

▶ **Lemma B.18** (Natural soundness). *If* $\Phi \to_{❀} \Psi$ *then* $\Phi \dashv\vDash \Psi$.

Proof. By Lemma B.17 and Lemma B.10. ◀

The soundness of cultural rules is straightforward with the previous lemmas:

▶ **Lemma B.19** (Cultural soundness). *If* $\Phi \to_{✂} \Psi$ *then* $\Psi \vDash \Phi$.

Proof. By inspection of every ✂-rule.

grow, crop By Lemma B.11 and Lemma B.10.
pull, glue By Lemma B.12 and Lemma B.10.
apis By Lemma B.15 and Lemma B.10.
apet By Lemma B.16 and Lemma B.10.

◀

Then it follows that every derivation in the flower calculus is sound:

▶ **Theorem B.20.** *If* $\Phi \to^* \Psi$ *then* $\Psi \vDash \Phi$.

Proof. By Lemma B.18, Lemma B.19 and Observation B.5, Observation B.6. ◀

In particular $\vdash \phi$ implies $\vDash \phi$, i.e. every provable flower is true.

C Detailed proofs

C.1 Deduction theorems

▶ **Lemma C.1** (Positive closure). *If* $\Phi \to \Psi$, *then* $\hat{\Xi}^+\{\Phi\} \to \hat{\Xi}^+\{\Psi\}$.

Proof. In the case of a natural step $\Phi \to_{❀} \Psi$, this is immediate by contextual closure (Definition 6.5). Otherwise we have a cultural step $\hat{\Xi}'\{\Phi_0\} \to_{✂} \hat{\Xi}'\{\Psi_0\}$. Then either $\hat{\Xi}'$ is positive, and $\mathsf{inv}(\hat{\Xi}^+\{\hat{\Xi}'\}) = \mathsf{inv}(\hat{\Xi}^+) + \mathsf{inv}(\hat{\Xi}')$ is even since it is the sum of two even numbers; or $\hat{\Xi}'$ is negative, and $\mathsf{inv}(\hat{\Xi}^+\{\hat{\Xi}'\})$ is odd since it is the sum of an even and an odd number. In both cases $\hat{\Xi}^+\{\hat{\Xi}'\}$ has the same polarity as $\hat{\Xi}'$, and thus the same rule can be applied. ◀

C.1.1 Proof of Theorem 6.7

Proof. Suppose that $\Phi \to^* \Psi$. Then we have:

$$\begin{array}{llll}
\Psi \supset \Phi & \to^* & \Psi \supset \Psi & \text{(Hypothesis + Lemma C.1)} \\
 & \to_{\mathsf{poll}\downarrow} & \Psi \supset \cdot & \\
 & \to_{\mathsf{epet}} & \varnothing &
\end{array}$$

In the other direction, suppose that $\Psi \,\mathsf{D}\, \Phi \to^* \varnothing$. Then we have:

$$\begin{array}{lll}
\Phi & \to_{\mathsf{epis}} & \mathsf{D}\, \Phi \\
& \to_{\mathsf{grow}} & (\Psi \,\mathsf{D}\, \Phi), (\mathsf{D}\, \Phi) \\
& \to_{\mathsf{poll}\uparrow} & (\Psi \,\mathsf{D}\, \Phi), ((\Psi \,\mathsf{D}\, \Phi) \,\mathsf{D}\, \Phi) \\
& \to^* & (\Psi \,\mathsf{D}\, \Phi) \,\mathsf{D}\, \Phi \qquad\qquad \text{(Hypothesis + Lemma C.1)} \\
& \to_{\mathsf{grow}} & \Psi, ((\Psi \,\mathsf{D}\, \Phi) \,\mathsf{D}\, \Phi) \\
& \to_{\mathsf{poll}\downarrow} & \Psi, ((\mathsf{D}\, \Phi) \,\mathsf{D}\, \Phi) \\
& \to_{\mathsf{srep}} & \Psi, (\mathsf{D}\, (\Phi \,\mathsf{D}\, \Phi)) \\
& \to_{\mathsf{poll}\downarrow} & \Psi, (\mathsf{D}\, (\Phi \,\mathsf{D}\, \cdot)) \\
& \to_{\mathsf{epet}} & \Psi, (\mathsf{D}\, \cdot) \\
& \to_{\mathsf{epet}} & \Psi
\end{array}$$

◀

C.1.2 Proof of Theorem 6.9

Proof. Let $\hat{\Xi}$ be some context. If $\Psi \vdash_{\text{❀}} \Phi$, then in particular $\hat{\Xi}'\{\Phi\} \to^*_{\text{❀}} \hat{\Xi}'\{\}$ for $\hat{\Xi}' \coloneqq \hat{\Xi}\{\Psi \,\mathsf{D}\, \square\}$. Thus we have:

$$\hat{\Xi}\{\Psi \,\mathsf{D}\, \Phi\} \to^*_{\text{❀}} \hat{\Xi}\{\Psi \,\mathsf{D}\, \cdot\,\} \to_{\mathsf{epet}} \hat{\Xi}\{\}$$

In the other direction, let $\hat{\Xi}$ be some context such that $\Psi \succ \hat{\Xi}$. If $\vdash_{\text{❀}} \Psi \,\mathsf{D}\, \Phi$, then in particular $\hat{\Xi}\{\Psi \,\mathsf{D}\, \Phi\} \to^*_{\text{❀}} \hat{\Xi}\{\}$. Thus we have:

$$\hat{\Xi}\{\Phi\} \to_{\mathsf{epis}} \hat{\Xi}\{\,\mathsf{D}\, \Phi\} \to_{\mathsf{poll}\uparrow} \hat{\Xi}\{\Psi \,\mathsf{D}\, \Phi\} \to^*_{\text{❀}} \hat{\Xi}\{\}$$

◀

C.2 Completeness

▶ **Lemma C.2** (Reflexivity). *For any bouquet Φ, $\Phi \vdash_{\text{❀}} \Phi$.*

Proof. Trivial by application of the $\mathsf{poll}\downarrow$ rule. ◀

▶ **Lemma C.3** (Weakening). *If $\mathcal{T} \subseteq \mathcal{T}'$ and $\mathcal{T} \vdash \phi$, then $\mathcal{T}' \vdash \phi$.*

Proof. This follows immediately from our definition of provability from a theory (Definition 8.1). ◀

In the following, we suppose some enumeration $(\phi_n)_{n\in\mathbb{N}}$ of $\mathbb{F}$, which should be definable constructively given the inductive nature of flowers. Let $\psi \in \mathbb{F}$, and $\mathcal{T}$ a ψ-consistent theory. We now define the *completion procedure*, which constructs an extension $\mathsf{Com}(\mathcal{T}) \supseteq \mathcal{T}$ with the property that $\mathsf{Com}(\mathcal{T})$ is ψ-consistent and ψ-complete.

▶ **Definition C.4** (n-completion). *The n-completion $\mathsf{Com}^n(\mathcal{T})$ of $\mathcal{T}$ is defined recursively by:*

$$\mathsf{Com}^0(\mathcal{T}) = \mathcal{T} \qquad \mathsf{Com}^{n+1}(\mathcal{T}) = \begin{cases} \mathsf{Com}^n(\mathcal{T}) \cup \phi_n & \text{if } \mathsf{Com}^n(\mathcal{T}) \cup \phi_n \text{ is } \psi\text{-consistent} \\ \mathsf{Com}^n(\mathcal{T}) & \text{otherwise} \end{cases}$$

▶ **Definition C.5** (Completion). *The completion $\mathsf{Com}(\mathcal{T})$ of $\mathcal{T}$ is the denumerable union of all n-completions:*

$$\mathsf{Com}(\mathcal{T}) = \bigcup_{n\in\mathbb{N}} \mathsf{Com}^n(\mathcal{T})$$

▶ **Lemma C.6.** *For every* $\psi \in \mathbb{F}$, $\mathsf{Com}(\mathcal{T})$ *is* ψ*-consistent and* ψ*-complete.*

Proof. It is immediate by induction on n that $\mathsf{Com}^n(\mathcal{T})$ is ψ-consistent. Then suppose that $\mathsf{Com}(\mathcal{T}) \vdash_{❀} \psi$, that is there is some bouquet $\Phi \subseteq \mathsf{Com}(\mathcal{T})$ such that $\Phi \vdash_{❀} \psi$. For each $\phi \in \Phi$, there is some rank n such that $\phi \in \mathsf{Com}^n(\mathcal{T})$. Let m be the greatest such rank. Then $\Phi \subseteq \mathsf{Com}^m(\mathcal{T})$, and thus by weakening (Lemma C.3) $\Phi \nvdash_{❀} \psi$. Contradiction. ◀

C.2.1 Proof of Proposition 8.3

Proof. Suppose the contrary, i.e. there is a substitution σ such that $\mathcal{T} \vdash_{❀} \sigma(\Phi)$ and for all $1 \leq i \leq n$, there is some $\phi_i \in \Phi_i$ ① such that $\sigma(\phi_i) \notin \mathcal{T}$. Thus by ψ-completeness of $\mathcal{T}$, we get $\mathcal{T}, \sigma(\phi_i) \vdash_{❀} \psi$. So there are $\Psi \subseteq \mathcal{T}$ and $\Psi_i \subseteq \mathcal{T} \cup \sigma(\phi_i)$ such that $\Psi \vdash_{❀} \sigma(\Phi)$ ② and $\Psi_i \vdash_{❀} \psi$ ③. Now it cannot be the case that $\Psi_i \subseteq \mathcal{T}$, otherwise by weakening and ψ-consistency of $\mathcal{T}$ we would have $\Psi_i \nvdash_{❀} \psi$. So there must exist $\Psi'_i \subseteq \mathcal{T}$ such that $\Psi_i = \Psi'_i \cup \sigma(\phi_i)$ ④. Again by weakening and ψ-consistency of $\mathcal{T}$, we get $\Psi, \bigcup_{i=1}^n \Psi'_i, \phi \nvdash_{❀} \psi$. Now we derive a contradiction by showing $\Psi, \bigcup_{i=1}^n \Psi'_i, \phi \vdash_{❀} \psi$. Let $\hat{\Xi}$ be a context such that $\Psi, \bigcup_{i=1}^n \Psi'_i, \phi \succ \hat{\Xi}$ ⑤. Then $\hat{\Xi}\{\psi\} \rightarrow^*_{❀} \hat{\Xi}\{\}$ with the following derivation:

$$
\begin{array}{llll}
\hat{\Xi}\{\psi\} & \rightarrow_{\mathsf{epis}} & \hat{\Xi}\{\cdot \;\mathsf{D}\; \cdot \psi\} & \\
 & \rightarrow_{\mathsf{poll}\uparrow} & \hat{\Xi}\{\cdot \phi \;\mathsf{D}\; \cdot \psi\} & (⑤) \\
 & \rightarrow_{\mathsf{ipis}} & \hat{\Xi}\{\cdot (\cdot \sigma(\Phi) \;\mathsf{D}\; \sigma(\Delta)), \phi \;\mathsf{D}\; \cdot \psi\} & \\
 & \rightarrow_{\mathsf{poll}\downarrow} & \hat{\Xi}\{\cdot (\cdot \;\mathsf{D}\; \sigma(\Delta)), \phi \;\mathsf{D}\; \cdot \psi\} & (②, ⑤) \\
 & \rightarrow_{\mathsf{srep}} & \hat{\Xi}\{\cdot \phi \;\mathsf{D}\; \cdot \{\sigma(\delta_i) \;\mathsf{D}\; \cdot \psi\}_i^n\} & \\
 & = & \hat{\Xi}\{\cdot \phi \;\mathsf{D}\; \cdot \{\mathbf{x}_i \cdot \sigma(\Phi_i) \;\mathsf{D}\; \cdot \psi\}_i^n\} & \\
 & \rightarrow^n_{\mathsf{poll}\downarrow} & \hat{\Xi}\{\cdot \phi \;\mathsf{D}\; \cdot \{\mathbf{x}_i \cdot \sigma(\Phi_i) \;\mathsf{D}\; \cdot\}_i^n\} & (①, ③, ④, ⑤) \\
 & \rightarrow^n_{\mathsf{epet}} & \hat{\Xi}\{\cdot \phi \;\mathsf{D}\; \cdot\} & \\
 & \rightarrow_{\mathsf{epet}} & \hat{\Xi}\{\} &
\end{array}
$$

◀

C.2.2 Proof of Proposition 8.4

Proof. Suppose the contrary, i.e. there are some $1 \leq i \leq n$ and $\sigma : \mathbf{x}_i$ such that $\mathcal{T}, \Phi \vdash_{❀} \sigma(\Phi_i)$. Therefore there must exist $\Psi \subseteq \mathcal{T}$ and $\Phi_0 \subseteq \Phi$ ① such that $\Psi, \Phi_0 \vdash_{❀} \sigma(\Phi_i)$ ②. By hypothesis, for every $\Phi' \subseteq \mathcal{T}$ there is a context $\hat{\Xi}$ such that $\Phi' \succ \hat{\Xi}$ and $\hat{\Xi}\{\phi\} \nrightarrow^*_{❀} \hat{\Xi}\{\}$. We now derive a contradiction by showing $\hat{\Xi}\{\phi\} \rightarrow^*_{❀} \hat{\Xi}\{\}$ for all $\hat{\Xi}$ such that $\Psi \succ \hat{\Xi}$ ③:

$$
\begin{array}{llll}
\hat{\Xi}\{\phi\} & \rightarrow_{\mathsf{ipet}} & \hat{\Xi}\{\mathbf{x} \cdot \Phi \;\mathsf{D}\; \cdot \sigma(\Phi_i) ; \Delta\} & \\
 & \rightarrow_{\mathsf{poll}\downarrow} & \hat{\Xi}\{\mathbf{x} \cdot \Phi \;\mathsf{D}\; \cdot ; \Delta\} & (①, ②, ③) \\
 & \rightarrow_{\mathsf{epet}} & \hat{\Xi}\{\} &
\end{array}
$$

◀

C.2.3 Proof of Lemma 8.6

Proof. By induction on $|\phi|$.

- Suppose $\phi = p(\vec{\mathbf{x}})$.
  1. By definition of forcing (Definition 7.5) and ✿(ψ) (Definition 8.5), $\mathcal{T} \Vdash p(\vec{\mathbf{x}})\,[\sigma]$ precisely when $\sigma(p(\vec{\mathbf{x}})) \in \mathcal{T}$.

2. Suppose that $\mathcal{T} \Vdash \phi\,[\sigma]$, that is $\sigma(\phi) \in \mathcal{T}$. Then by weakening (Lemma C.3), we get $\sigma(\phi) \nvdash_{\circledast} \sigma(\phi)$. But this is impossible by reflexivity of $\vdash$ (Lemma C.2).

- Suppose $\phi = \mathbf{x} \cdot \Phi \supset \{\mathbf{x}_i \cdot \Phi_i\}_i^n$.
  1. Let $\mathcal{U} \supseteq \mathcal{T}$ be a ψ-consistent and ψ-complete theory. Obviously $\sigma(\phi) = \mathbf{x} \cdot \sigma|_{\mathbf{x}}(\Phi) \supset \{\mathbf{x}_i \cdot \sigma|_{\mathbf{x} \cup \mathbf{x}_i}(\Phi_i)\}_i^n \in \mathcal{U}$, and thus by Proposition 8.3, for every substitution τ, either $|_{\mathbf{x}}\tau \circ \sigma|_{\mathbf{x} \cup \mathbf{x}_i}(\Phi_i) = \sigma|_{\mathbf{x}}\tau|_{\mathbf{x}_i}(\Phi_i) \subseteq \mathcal{U}$ for some $1 \leq i \leq n$, or $\mathbf{U} \nvdash_{\circledast} |_{\mathbf{x}}\tau \circ \sigma|_{\mathbf{x}}(\Phi) = \sigma|_{\mathbf{x}}\tau(\Phi)$. In the first case, we get $\mathcal{U} \Vdash \Phi_i\,[\sigma|_{\mathbf{x}}\tau|_{\mathbf{x}_i}]$ by IH. In the second case, we get $\mathcal{U} \nVdash \Phi\,[\sigma|_{\mathbf{x}}\tau]$ by IH. In other words, $\mathcal{U} \Vdash \Phi\,[\sigma|_{\mathbf{x}}\tau]$ implies $\mathcal{U} \Vdash \Phi_i\,[\sigma\,|_{\mathbf{x}}\,\tau\,|_{\mathbf{x}_i}]$, that is $\mathcal{T} \Vdash \phi\,[\sigma]$.
  2. By Proposition 8.4, for every $1 \leq i \leq n$ and substitution τ, there is some $\phi_i \in \Phi_i$ such that $\mathbf{T}, \sigma|_{\mathbf{x}}(\Phi) \nvdash_{\circledast} |_{\mathbf{x}_i}\,\tau \circ \sigma|_{\mathbf{x} \cup \mathbf{x}_i}(\phi_i) = \sigma\,|_{\mathbf{x}}\,|_{\mathbf{x}_i}\,\tau(\phi_i)$. By the completion procedure, we get a theory $\mathcal{U} = \mathsf{Com}(\mathcal{T} \cup \sigma|_{\mathbf{x}}(\Phi)) \supseteq \mathcal{T} \cup \sigma|_{\mathbf{x}}(\Phi)$ which is both $\sigma\,|_{\mathbf{x}}\,|_{\mathbf{x}_i}\,\tau(\phi_i)$-consistent and $\sigma\,|_{\mathbf{x}}\,|_{\mathbf{x}_i}\,\tau(\phi_i)$-complete. Then by IH, $\mathcal{U} \Vdash \Phi\,[\sigma\,|_{\mathbf{x}}\,]$ since $\sigma\,|_{\mathbf{x}}\,(\Phi) \subseteq \mathcal{U}$, and $\mathcal{U} \nVdash \phi_i\,[\sigma\,|_{\mathbf{x}}\,|_{\mathbf{x}_i}\,\tau]$ since $\mathcal{U}$ is $\sigma\,|_{\mathbf{x}}\,|_{\mathbf{x}_i}\,\tau(\phi_i)$-consistent, that is $\mathcal{T} \nVdash \phi\,[\sigma]$.

◀

C.2.4 Proof of Theorem 8.7

Proof. Let $\mathcal{T}$ be a ψ-consistent theory. We prove that $\mathcal{T} \nvDash \psi$ by showing in particular that $\mathcal{T} \nvDash_{\circledast(\psi)} \psi$, and more specifically that $\mathsf{Com}(\mathcal{T}) \Vdash \mathcal{T}\,[1]$ but $\mathsf{Com}(\mathcal{T}) \nVdash \psi\,[1]$. Then it follows by (classical) contraposition that $\mathcal{T} \vDash \psi$ implies $\mathcal{T} \vdash_{\circledast} \psi$ for any ψ and any $\mathcal{T}$, and thus we can conclude.

- Let $\phi \in \mathcal{T}$. Then $1(\phi) = \phi \in \mathsf{Com}(\mathcal{T})$, thus by ψ-consistency and ψ-completeness of the completion (Lemma C.6), one can apply adequacy (Lemma 8.6) to get $\mathsf{Com}(\mathcal{T}) \Vdash \phi\,[1]$.
- Similarly, we can apply adequacy (Lemma 8.6) to get $\mathsf{Com}(\mathcal{T}) \nVdash \psi\,[1]$.

◀