Exceptions for algebraic specifications: on the meaning of “but”

Pierre-Yves Schobbens
CNRS, CRIN, BP 239, F-54506 Vandœuvre-lès-Nancy, France

Abstract

When building large specifications from requirements, the structure of the specification becomes a central problem: the specification language should allow a decomposition that closely reflects the structure of requirements. In this paper, we propose a decomposition into defaults (general rules) and exceptions to these general rules that fits the requirements found in some application domains. It is complementary, and builds upon, the modular decomposition proposed by the algebraic specification school. Its definition is based on abstract model theory, leading to the definition of default institutions.

Keywords. Exceptions; nonmonotonic logic; algebraic specifications.

1. Introduction

Large specifications need structure to be manageable. This structure is key to their readability, writability, reusability, and traceability to requirements. The algebraic specification school proposes a strong structuring construct, importation, where an existing specification can be enriched, but not be modified. While developing case studies [14, 15], we discovered that this structure is often, but not always, adequate:

- It might not match the organization found in requirements, making it difficult to relate specifications to the original requirements.
- It might not reflect the construction process followed by the specifier, leading to difficulties in explaining the specification, and sometimes to large modifications in the specification in response to a small change in requirements.
It might hinder reuse, since existing specifications often have to be slightly adapted.

We propose here a complementary structure that allows partial reuse of existing specification modules: the "default" module is used as a template that can be tailored by introducing specific exceptions to suit the needs of the application at hand. To give a meaning to such a combination, we select the models of the exception that are as close as possible to the models of the default. The use of models preserves the level of abstraction found in algebraic specifications: specially, logically equivalent specifications are not distinguished.

This composition is nonmonotonic: by adding exceptions, we may have to retract defaults. Note that nonmonotonicity is already present in the initial approach to algebraic specifications: in the initial model, all terms are distinguished by default, and by adding equalities we may have to retract some of these default inequalities. In the same way that the initial approach can be parameterized by the logic used and the morphisms between models (that are used to select the initial model) [24], our proposal can be parameterized by the logic and the closeness relation between models, leading us to define default institutions.

To allow the reader to become acquainted progressively with the generality of default institutions, we start with the simplified case of a default $D$ and an exception $E$, that we note $D$ but $E$. A propositional logic is first presented informally (Section 3.1.1); it is then generalized to a first variant of predicate logic (Section 3.1.2), but this variant cannot deal with exceptions on equality. A better variant is thus designed, and at the same time we show how it can be abstracted to default institutions (Section 3.2.1). Using default institutions, the general case with several defaults is then presented in Section 4, under the name of reliability graphs. Sound rules for a logic of reliability graphs are presented.

1.1. Examples of application

(1) Specification construction: Even when requirements are well understood, the construction of a formal specification is incremental, beginning with a simplified model of the system to be built, while exceptions to this model are introduced at a later stage. For instance, in the specification of the Unix System V file system, the `mv` command has a simple, orthogonal definition [14, Fig. 5], but some cases (e.g., moving a directory) must produce an error message instead. These cases are contradicted by further exceptions (e.g., moving a directory onto a file) that must produce more specific error messages.

(2) Requirements elicitation: When integrating requirements from different users, contradictions often occur that are solved through the use of a precedence between requirements [48]. The present scheme can be used to solve simple problems of this type.
(3) Specification of domains involving exceptions:

(a) Grammar: The grammar of most natural languages is presented through general rules later contradicted by special cases, and recursively. For instance, in French, verbs whose infinitive ends in “er” form their third plural of present indicative through the suffix “ent”, but verbs in “eler” will use “ellent”, except some that use “élent”.

(b) Biological taxonomy: Animals are grouped into classes described by prototypical properties, but many subclasses are defined through exceptions to these properties.

In each case, exceptions could be eliminated in the final specification, at the expense of readability, writability, and traceability.

2. Institutions

The basic definitions of the algebraic school (who borrowed them from logic and algebra) are recalled below, with an emphasis on structuring concepts.

They are independent of the underlying logic. The usual requirements on this logic (called an institution [10, 24]) are introduced as we go along, and will be extended in the next section towards default institutions, by adding a closeness between interpretations that will be used to combine defaults. In fact, as these requirements are rather abstract, most of our efforts will be devoted to their illustration on a simple, yet not trivial, variant of first-order logic that has interesting properties for the specification and implementation of abstract data types. This section first recalls the syntactical and semantical parts of institutions. Institutions can then be used to define generic operations on specifications: initiality (or freeness) [24], importation, inclusion, and reachability.

2.1. Syntax: modules

A specification is a description, or better a modeling of the reality. This description uses a language. In fact part of this language is chosen by the user; we say that the language is parameterized by a signature, and we denote it by $\mathcal{L}(\Sigma)$.

A presentation is composed of a signature $\Sigma$ and of a class of formulae (called its axioms) written in $\mathcal{L}(\Sigma)$.

Example 2.1. Our running example will be surjective multi-sorted first-order logic without equality that we abbreviate SFOL. The language of SFOL is formed on basis of symbols declared in a signature $\Sigma$, which contains:

- $S$, a set of sorts;
- $O$, a set of operators with functions $\alpha : O \rightarrow S^*$ giving the sort of their arguments and $\sigma : O \times S$, the sort of their result;
- $P$, a set of predicates with $\alpha : P \rightarrow S^*$ as above.

The signature is used to build the usual first-order language.
Example 2.2. Our concrete examples (that show the use of our example institution) borrow their syntax from LPG [2,3], and their graphical notation from Z [52], where a module is represented by a box with a line dividing the signature from the axioms.

The natural numbers are specified in Fig. 1 as an example of a data structure definition typical of software specification.

2.2. Semantics: models

The semantics of a module will be defined here as its class of models. From an abstract point of view, we just need a class of interpretations \( M(\Sigma) \) for each signature, and a satisfaction relation \( \models \in M(\Sigma) \times \Sigma(\Sigma) \).

Example 2.3. An algebra \( A \) of a signature \( \Sigma \) gives
- for each sort \( s \), a set \( s_A \) (called the carrier of the sort);
- for each operator symbol \( f \), a function \( f_A \) from the carriers of the argument sorts to the carrier of the result sort;
- for each predicate symbol \( p \), a relation \( p_A \) between the carriers of the argument sorts.

In SFOL, we assume that interpretations have an internal signature \( \Sigma A \) containing \( \Sigma \); an interpretation is thus a pair \( (\Sigma A, A) \), where \( A \) a surjective \( \Sigma A \)-algebra. This definition will later be useful to give a good structure to correspondences (see Theorem 2).

A valuation \( V \) is a function that for each variable yields its value, i.e. a member of the carrier of its sort. We say that \( V' \equiv \approx V \), if \( \forall x \in X, \{v\}, V'(X) = V(X) \).

The evaluation \( V_A \) is the function that extends \( V \) by assigning to each term of \( T_{\Sigma A}(X) \) a value (an element of the carrier of its sort) so that \( V_A(f(t_1, \ldots, t_n)) = \)
On the meaning of “but”

For any operator $f \in \Sigma A$. For each valuation, there is exactly one evaluation extending it. When no variables are present ($X = \emptyset$), there is a single ground valuation, which is the empty function. The ground evaluation, denoted $e_A$, is the corresponding evaluation that gives a value to each (internal) ground term. If $e_A$ is injective, the algebra is called injective as well; if $e_A$ is surjective, the algebra is called surjective; if the restriction of $e_A$ to the ground terms of the signature $\Sigma$ is surjective, the algebra is called term-generated. Surjective algebras have a special importance in computer science, due to their constructive nature [11, 57]. We follow this tradition and define interpretations of SFOL as surjective algebras. As the signature of the algebra can be larger than the signature of the language, this is not a real restriction, since we can always add constants to represent each element (i.e. form diagram signature [1]), but it will have its importance when we consider the closeness between interpretations. An algebra $A$ satisfies a formula $\phi$ for a valuation $V$, denoted $A \models \_ (\phi)$, if:

- $A \models p(t_1, \ldots, t_n)$ iff $p_A(V_A(t_1), \ldots, V_A(t_n))$;
- $A \models \phi_1 \land \phi_2$ iff $A \models \phi_1$ and $A \models \phi_2$;
- $A \models \neg \phi$, iff $A \models \phi$ is false;
- $A \models \forall v, \phi$ iff $A \models \phi$ for all $V' \approx V$.

An algebra $A$ satisfies a formula $\phi$, noted $A \models \phi$, if it satisfies it for all valuations.

A model of a presentation $P$ is an interpretation of the signature of $P$ that satisfies the axioms of $P$. The semantics of a presentation $P$ is the class of its models, denoted $\text{Mod}(P)$.

Let’s see what happens when the signature varies. The signatures should form a category, with morphisms showing variations. A signature morphism $m : \Sigma \to \Sigma'$ should induce a translation function $\text{Tr}_m$ on the language and an opposite forgetful functor $\text{lm}$ on algebras. We say that:

- A formula $\phi \in \mathcal{L}(\Sigma)$ entails $\phi' \in \mathcal{L}(\Sigma')$, denoted $\models \phi \Rightarrow \phi'$, iff all models of $\phi$ are models of $\phi'$. This induces a category of formulae.
- A presentation $P$ entails a formula $\phi \in \mathcal{L}(\Sigma P)$, denoted $P \models \phi$, iff all models of $P$ satisfy $\phi$. We also say that $\phi$ is a property of $P$.
- A theory is an entailment-closed class of formulae: $\phi \in T$, $\phi \models \psi \Rightarrow \psi \in T$. (Theories equipped with inclusion form yet another category.) Specially, the properties of a presentation form its closure theory $\text{Cl}(P)$. Similarly, the theory $\text{Cl}(\Sigma, M)$ of a model $M$ is the set of its properties: $\{\phi \in \mathcal{L}(\Sigma) \mid M \models \phi\}$.

We define semantic equivalence $\equiv$ as isomorphism in each of these categories.

A model $M$ is refined by another model $N$, $M \sim N$, if $N$ has more properties: $\text{Cl}(\Sigma, M) \subseteq \text{Cl}(\Sigma, N)$. On the other hand, we have assumed (following the tradition of institutions) that interpretations of a given signature form a category. The morphisms between interpretations are usually variants of homomorphisms (see [24] for examples), but we will see in Definition 1 another definition that suits better
our purposes. These morphisms naturally define another preorder, and it is instructive to compare the two preorders. The institution is said to be weakly abstract iff \( M \sim N \) whenever \( \exists h : M \Rightarrow N \). Intuitively, an institution is weakly abstract if our logic does not allow us to look at more details of our models than the morphisms do.

**Example 2.4.** First-order logic with equality, equipped with isomorphisms, is weakly abstract. (end of 2.4)

**Example 2.5.** Branching temporal logic with bisimulations [18] is weakly abstract. (end of 2.5)

**Example 2.6.** SFOL is weakly abstract (see Theorem 3). (end of 2.6)

2.3. Initial model

Institutions are used in [24] to define initial models: a model is initial if there is a single morphism from it to any other model. Informally, morphisms define an order of preference, and we choose the unique minimum (if it exists) as the semantics of our specification. In the usual equational institution [17], this model has the interesting properties of no junk and no confusion (see Section 5.4.1). The language designer can construct morphisms to select the model according to his taste. For instance, in partial algebras, Möller [40] and Reichel [45] choose the least defined model. We think that this concept can be improved in two main directions:

- Requiring the existence of a minimum is too rigid. The language is then restricted to Horn clauses, excluding thus disjunction or existential quantification that we consider to be important tools for specification. We will therefore work with minimal models instead (defined in Section 3.3).
- Although institutions allow the language designer to choose his criteria for minimization, the language user (the specifier) has no choice. Here, we will allow the specifier to give an “ideal template”, also called a default. The models that are closest to this default will be retained.

3. Default institutions

3.1. Examples

3.1.1. Propositional logic

We first present the case of classical propositional logic. This allows a simple and intuitive exposition of the basic structure of default institutions.

What should be the meaning of but in this simple setting? Clearly, when no conflicts arise between the default and the exception, we should just gather the information by a simple conjunction. For instance, given a signature with two propositions \( J \) for “JR comes at the party” and \( S \) for “Sue Ellen comes at the
party”, we intuitively understand “JR comes at the party, but not Sue Ellen” as $J \land \lnot S$.

If conflicts arise, we would like to keep as much information from the default as possible. For instance, from “Bobby, JR, and Sue Ellen come at parties, but Sue Ellen is not coming tonight”, $(B \land J \land S)$ but $(\lnot S)$ means, intuitively, $B \land J \land \lnot S$.

To reach the abstraction level usual in algebraic specifications, and to avoid any bias given by the concrete syntax of formulae, we would like our formalization to be based on models. A Karnaugh table (Fig. 2) can help us to visualize the models: in such a table, each entry represents a possible model.

![Karnaugh table of the models.](image)

Looking at the table, the models we want are the models of $E$ that are closest to a model of $D$. Here, two such models exist: $e_1$ described by $B \land J \land \lnot S \land P$ (which is $S$ away from $d_1 = B \land J \land S \land P$) and $e_2$ described by $B \land J \land \lnot S \land \lnot P$ (which is also $S$ away from $B \land J \land S \land \lnot P$).

This institution can easily be formalized: Given two models $M$ and $N$ we can define the closeness between $M$ and $N$, denoted $d(M, N)$, as a pair of sets of propositions:

- the first set contains the propositions that are true in $M$ and false in $N$;
- the second set contains the propositions that are false in $M$ and true in $N$.

The distances will be compared by componentwise inclusion. Propositional logic equipped with this closeness will be called $PL$. It is equivalent to the propositional fragment of the default institutions presented in the sequel. Its strong intuitive appeal stems, we think, from the fact that the closeness between models is determined by the difference between their elementary facts. Of course, nothing ensures that real-world facts will correspond to logical facts (propositions), but it is a reasonable starting point. Unlike the usual concept of distance, the closeness used here:

- is not really symmetrical: $d(e_1, d_1) = (\emptyset, \{S\})$, while $d(d_1, e_1) = (\{S\}, \emptyset)$.
- is not totally ordered: for instance $d(d_1, d_2) = (\{P\}, \emptyset)$, while $d(e_4, e_2) = (\emptyset, \{J\})$, and these are not comparable by componentwise set inclusion.

Of course, other closenesses are possible:

- To ensure symmetry, we could just take the union of the two components of the pair, so that for instance $d(e_1, d_1) = \{S\}$. In this case, we consider that suppressing or adding a fact amounts to the same change. This closeness is proposed in [56].
• To ensure linearity (total ordering), we can consider the size of the sets in either of the two proposals above. Among other bad properties, this closeness would be sensitive to the duplication of propositions: if we use two propositions $p$ and $p'$ to represent the same information, the weight of this information will increase in the closeness. This closeness is proposed in [13].

• This last inconvenience can be avoided by using user-defined weights.

Our study will be parameterized by the logic and by the closeness chosen. This flexibility can be used to introduce domain-dependent closenesses favoring for instance one proposition over another, according to their contents; but we prefer to work in a single default institution, and to obtain this type of preferences by other means, as done in Section 4. Algorithms and proof rules can then be devised once and for all.

3.1.2. Predicate logic (circumscription)

We would like now to introduce variables in our previous proposal (PL).

A natural idea is to replace the propositions $p$ by $p(a_1, \ldots, a_n)$, where $a_1, \ldots, a_n$ are elements of the carriers. This leads to the definition of a closeness (called $FOL=\equiv$) where each predicate is mapped to the differences between its extensions in the two interpretations: $d(M, N) = \{p \mapsto (p_M \setminus p_N, p_N \setminus p_M)\}$. This closeness only has a meaning if the carriers of $M$ and $N$ are the same; furthermore, the functions should also be the same in the two models. This example shows that a further generalization of the concept of distance is needed: the closeness, instead of being a total function, might be defined only for some pairs of models (here, models that have the same carriers and functions).

One consequence of this definition is that exceptions on equality cannot be dealt with: for instance, if two constants are requested to be equal in the default and different in the exception, then some carrier or some function has to be different. Even if we convene that equality is not built in, the same problem occurs (this is well known in the study of circumscription, see [20]). This is rather unfortunate as most existing algebraic specifications make heavy use of equality.

3.1.3. The problem of equality

In the framework of algebraic specifications, models with different carriers are usually compared using homomorphisms. Between two models, there may exist several homomorphisms, so that a further generalization will be needed: the closeness may depend not only on the models, but also on the way they are compared. We will thus replace pairs $(M, N)$ by morphisms $h : M \rightarrow N$ that may contain supplementary information—typically information about which element in a carrier of one model is the counterpart of a given element in the corresponding carrier of the other model. Partiality is then represented naturally by the absence of morphisms between the two models considered. In the previous example ($FOL=\equiv$), the morphisms are equalities. Given an homomorphism $h : A \rightarrow B$, the closeness will thus be generalized
to a mapping giving for each predicate \( p : w \), the pair
\[
\{(\bar{a} \in w_A | p_A(\bar{a})) \text{ and not } p_B(h(\bar{a}))\},
\]
\[
\{\bar{a} \in w_A | \text{not } p_A(\bar{a}) \text{ and } p_B(\bar{a})\}\}.
\]

We are led to compare two homomorphisms by using a third one: \( h : A \to B \leq h' : A' \to B' \) iff \( \exists h_A : A \to A'; \forall p : w \in P; \forall \bar{a} \in w_A: \)

- \((p_A(\bar{a}) \text{ and not } p_B(h(\bar{a}))) \implies (p_A(h_A(\bar{a})) \text{ and not } p_B(h'(h_A(\bar{a}))))\),
- \((\text{not } p_A(\bar{a}) \text{ and } p_B(h(\bar{a}))) \implies (\text{not } p_A(h_A(\bar{a})) \text{ and } p_B(h'(h_A(\bar{a}))))\).

To simplify notation, we have suppressed the function \( d \), and we use a preorder between morphisms directly. In the special case where the default is of the form \( \wedge_{p : w \in P} \forall X : w; \neg p(X) \), we obtain equality circumscription [44]. This ordering introduces more problems than it solves:

**Example 3.1.** Consider the trivial specification \( D \textbf{ but } D \). It is intuitively expected to be equivalent to \( D \). And indeed, the measure of distance between a model and itself using the identity morphism is empty. The definition of \( h : A \to B \leq h' : A' \to B' \) given above simplifies thus to \( \exists h_A : A \to A' \), which will select a model where equality is minimal. We see thus that the use of a homomorphism in the comparison leads to:

- minimizing the equality even when nothing in the specification seems to request it;
- considering that some pairs of models might agree strictly better than a model with itself.

In the definition of default institution, we will take care to eliminate this counter-intuitive behaviour by requesting that identities are always minimum. (end of 3.1)

Furthermore, this ordering deals somewhat unexpectedly with elements that are not the value of a ground term:

**Example 3.2.** it is common knowledge that one seldom wins in games of chance, so let \( D = \forall x : \text{person}; \neg \text{wins}(x) \). We are told that Harry played heads and tails with an unknown person: \( E = \exists u : \text{person}; u \neq \text{Harry} \land (\text{wins}(u) \lor \text{wins}(\text{Harry})) \). By examining the models of \( D \textbf{ but } E \), we can deduce that Harry has lost the game. This unexpected answer disappear when we introduce a name for \( u \) (by skolemizing \( E \)). (end of 3.2)

**3.2. Default institutions**

Our examples had a double motivation:

- to show that the treatment of the examples required several generalizations of the concept of distance, and also to show which properties of the distance should be preserved;
to show that the treatment of exceptions on equality had to be improved, leading to the proposal of SFOL that will be presented in more detail as an example of a default institution.

3.2.1. Morphisms

Our first problem is thus to relate algebras that may be widely different in nature: for instance, natural numbers, strings of bits, counters of an abacus are all algebras of Nat. We need therefore a way to relate elements of different nature that play a similar role. In fact, classical institutions [10] already assume morphisms to that end.

Example 3.3. Let us return to the SFOL example. We use correspondences for the internal signature as morphisms between SFOL interpretations. (The term is borrowed from [6].)

Definition 1. Let $A$ and $B$ be two algebras of $\Sigma$. A correspondence for $\Sigma$ between $A$ and $B$ is a family of relations, $\sim_s$, between the carriers of $A$ and $B$, with the following properties:

1. compatible with operators: $\forall f: s_1, \ldots, s_n \to s \in O; \ a_1 \sim_a b_1, \ldots, a_n \sim_a b_n \Rightarrow f_A(a_1, \ldots, a_n) \sim f_B(b_1, \ldots, b_n)$,
2. total: $\forall a \in s_A, \exists b \in s_B, \ a \sim_b b$,
3. surjective: $\forall b \in s_B, \exists a \in s_A, \ a \sim_a b$.

Example 3.4. The natural numbers, $\mathbb{N}$, and the natural numbers modulo 3, $\mathbb{N}/3$, are algebras of $\Sigma$Nat. The following relations are correspondences between them:

1. $m \sim_1 n$ iff $m$ modulo 3 is $n$,
2. $\sim_2$ is the full relation $\mathbb{N} \times \mathbb{N}/3$.

The first is included in the second; we also say that the second is stronger, or coarser.

Theorem 1. Correspondences form a category for relational composition.

Theorem 2. The correspondences between two surjective algebras form a complete lattice.

3.2.2. Ordering

The last problem is to effectively compare morphisms (i.e., pairs of models linked by indications about which elements play similar roles), in order to give a precise meaning to "closest". To that end, we require a preorder $\preceq$ among morphisms.

We require that identity morphisms are minima. Consequently, all of them are equivalent for the ordering, and we convene to note one among them as $0$.

Let us say that $d$ is an agreement if $d = 0$. It is wanted that interpretations that agree (for instance, isomorphic interpretations) behave similarly in all respects. Therefore, we require agreements to form a subcategory $Int_0$ of $Int$ such that the resulting institution is weakly abstract: this ensures reflexivity, transitivity, and
On the meaning of "but"

indistinguishability for formulae of the original language. For the same reason, a
default institution should be \(\theta\)-symmetrical, that is: for each agreement \(h : A \to B\)
there is a reverse agreement \(h^R : B \to A\) such that \((h^R)^R = h\). Finally, agreements
should also be transparent with respect to comparison. This property is expressed
by \(\theta\)-equivalence: for any morphism \(h : B \to C\) and agreements \(a : A \to B, c : C \to D,\)
\(a;h = h = h;c\).

This closes our requirements on default institutions, summarized in Appendix A.1.
These requirements suggest a method for finding adequate categories:

1. start from an abstract category \(\mathcal{F}\); such off-the-shelf categories exist for most
   logics;
2. generalize it keeping the structural part only;
3. choose the comparison preorder such that \(d = 0\) iff \(d\) is a morphism of \(\mathcal{F}\).

This method has been followed for our example default institutions, except that we
have found no abstract category for logics without equality on our shelf.

**Example 3.5.** In SFOL, the structural part of morphisms relate constants and
functions, so that we choose to measure the difference between predicate extensions.
We first introduce a concrete representation for that difference.

**Definition 2.** A double correspondence from a correspondence \(\sim : A \to B\) to another
\(\sim' : A' \to B'\) is a family of relations \(\approx\) (indexed by sorts) between pairs of corresponding
elements, with the same properties as correspondences:

1. compatible with operators: \(\forall f : s_1, \ldots, s_n \to s \in \mathcal{O}; (a_1, b_1) \approx (a'_1, b'_1), \ldots,
   (a_n, b_n) \approx (a'_n, b'_n) \Rightarrow (f_\lambda(a_1, \ldots, a_n), f_\mu(b_1, \ldots, b_n)) \approx (f_\lambda(a'_1, \ldots, a'_n),
   f_\mu(b'_1, \ldots, b'_n));\)
2. total: \(\forall s \in S, \forall (a, b) \in \sim; \exists (a', b') \in \sim'; (a, b) \approx (a', b');\)
3. surjective: \(\forall s \in S, \forall (a', b') \in \sim'; \exists (a, b) \in \sim; (a, b) \approx (a', b').\)

**Definition 3.** The SFOL preorder on correspondences, noted \(\sim \preceq \sim'\), holds if there
is a double correspondence \(\approx : \sim \to \sim'\) such that any conflict between \(A\) and \(B\) has
a corresponding conflict between \(A'\) and \(B'\), i.e. for all \((a', b')\) such that \((a, b) =
(a', b'):\)

- \(p_\lambda(a) \land \neg p_\mu(b) \Rightarrow p_\lambda(a') \land \neg p_\mu(b');\)
- \(\neg p_\lambda(a) \land p_\mu(b) \Rightarrow p_\lambda(a') \land p_\mu(b').\)

**Theorem 3.** If two SFOL interpretations agree, they satisfy the same first-order
formulae.

3.3. The meaning of but

The models of \(D\) but \(E\) are the models of \(E\) that are the closest to a model of
\(D\); the comparison category chosen by the language designer provides the (general-
ized) distance that gives the meaning of "closest".
We introduce some notations to express this formally. For formulae \( D, E \in \mathcal{L} \), we define \( \text{Mor}(E, D) \) as the morphisms whose domain satisfy \( E \) and whose codomain satisfy \( D \).

**Definition 4.** \( h \) is *minimal* among a set of morphisms \( S \ (h \in \text{Min}(S)) \) iff:
- \( h \in S \);
- \( \forall h' \in S \), \( h' \leq h \Rightarrow h \leq h' \).

We abbreviate \( \text{Min} (\text{Mor}(E, D)) \) in \( \text{Min}(E, D) \), so that we can write:

**Definition 5.** \( e \models D \text{ but } E \) iff \( \exists h \in \text{Min}(E, D) \); \( e = \text{dom}(h) \).

Informally, a model of \( D \text{ but } E \) is the starting point of a shortest path from \( E \) to \( D \).

**Example 3.6.** The sentence "all men are mortal, but Faust" can be modelled as:

\[ \forall m: \text{man}, \text{mortal}(m) \text{ but } \neg \text{mortal}(\text{Faust}). \]

We expect intuitively that the disagreement between a model \( A \) of \( D \) and a model \( B \) of \( E \) are the immortal men of \( B \). The smallest such set will only contain Faust, so that in all models of \( D \text{ but } E \) all men but Faust will be mortal.

Many uses of exceptions in linguistics are of this type: a general rule admits a number of specific exceptions.

Let us look in more detail if the informal reasoning above is valid in all our example default institutions.
- In FOL\(-\), we can derive \( \forall x: \text{man}, x \not= \text{Faust} \Rightarrow \text{mortal}(x) \), but we cannot derive \( \text{mortal}(\text{John}) \)—assuming John is another constant of type man—unless we specify \( \text{John} \not= \text{Faust} \).
- In SFOL, Faust is the unique immortal man, as expected, and we can derive \( \text{mortal}(\text{John}) \).

**Example 3.7.** The sentence "All Mohicans are dead, but one" can be modelled as:

\[ \forall m: \text{Mohican}, \text{dead}(m) \text{ but } \exists m': \text{Mohican}, \neg \text{dead}(m'). \]

Once again, the disagreement will be the Mohicans that are not dead in the model of \( E \), and the smallest disagreements are the singletons, but here the models may disagree on who is the last surviving Mohican.

**Example 3.8.** Natural numbers modulo 3 are obtained as in Fig. 3.

This supplementary law \( (0 - 3) \) contradicts the first two axioms of Nat. In SFOL, the disagreement is given by the numbers that were different in the model of Nat and now become equal. The smallest disagreement is given by the natural numbers modulo 3, \( \mathbb{N}/3 \). Note that this example is plainly unsatisfiable in FOL\(=\).

(end of 3.8)
4. Reliability graphs

4.1. Definition

4.1.1. Syntax

It is often useful to be able to order not only two "sources of information", as with but, but an arbitrary number of them, with an arbitrary precedence relation. The same idea is found in [26, 31, 49, 55]. To that end we define a reliability graph for a given signature $\Sigma$ as:

- a set $S$ of "sources of information" (graphically represented by points);
- a well-founded partial order $<_S$ on $S$, where the lowest source is considered as the most reliable;
- a function $\Phi$ from $S$ to formulae, representing the knowledge provided by each source.

If $G$ is the name of the graph, we sometimes denote $\Phi(s)$ by $G_s$.

Example 4.1. See Fig. 4.

---

Fig. 3. Data structure definition of the natural numbers modulo 3.

Fig. 4. Example 4.1.
4.1.2. Lexicographic ordering

The definition of a model of such a graph is the obvious extension of the definition of but, where the concept of morphism is replaced by family of morphisms. The ordering among families is “lexicographic”:

\[ \vec{h} \leq_G \vec{h}' \iff \forall s \in S, h_s \leq h'_s \Rightarrow \exists t < s, h_t < h'_t. \]

More precisely, this definition extends the concept of lexicographic ordering in two directions: both the order of “string positions” (sources) and the order of “characters” (morphisms) can be partial. When the sources are not ordered at all, the ordering of families is the usual ordering of tuples. At the opposite, when the sources are linearly ordered, we obtain the usual concept of lexicographic ordering. This extended lexicographic ordering can be found in [26, 49], among others.

4.1.3. Semantics

We simply adapt the definition of but: \( Mor(G) \) is defined as \{ \vec{h} \in \{ e, d_i; h_i: e \to d_i, d_i \subseteq G_i \} \}. \( Min(G) \) contains the minimal morphisms of \( Mor(G) \), i.e. \{ \vec{h} \in Mor(G) \exists \vec{h}' \in Mor(G), \vec{h}' <_G \vec{h} \}.

Definition 6. \( e \models G \) iff \( \exists \vec{h} \in Min(G); \forall s \in S; dom(h_s) = e. \)

4.1.4. Operations on graphs

It is often useful to create new graphs by combining existing ones. We have isolated the following useful operations on graphs:

- The superposition of \( G \) over \( G' \), denoted \( G / G' \), places \( G \) above \( G' \), i.e. \( G \) is deemed less reliable than \( G' \). This operation can be used to integrate new information taking precedence.
- The juxtaposition of \( G \) and \( G' \), denoted \( G \parallel G' \), places \( G \) and \( G' \) side by side. This operation can be used to integrate new information in a skeptical way.
- The replacement of a source by a subgraph, \( G[s := G'] \), is a generalization of the previous operations: Juxtaposition is obtained by replacing the two nodes of a binary graph without precedence, and superposition by replacing the two nodes of a binary graph with a precedence link.
- The single-node operation takes a formula and makes a graph of it. This operation will be left implicit.
- The empty graph operation, \( 0 \), is included for theoretical purposes only.

We also say that a graph \( H \) has more edges than \( G \), if \( \leq_H \supseteq \leq_G \), and the sources and formulae are the same.

These operations extend naturally to families of morphisms. Note that these operations take into account the internal structure of graphs: Even if two graphs have the same class of models, one may not replace the other as the argument of such graphical operators. If we identify isomorphic graphs, we obtain that “/” is
 associative with identity $\emptyset$, and $\|\|$ is commutative and associative with identity $\emptyset$. Both are monotonic with respect to "has more edges".

For any graph $G$, the semantics of $G$, denoted $\overline{G}$ or $|G|$, defines a class of models, just like a formula. Framed graphs can thus be used in place of formulae: for instance, they may be combined by the usual boolean connectives. For the same reason, framed graphs may be used inside graphs, allowing a finer treatment of precedence. For instance, if we interpret Fig. 5 in the default institution $PL$, the first graph, that we can also note $p/\langle p\|\neg p\rangle$, means just "true": the upper source asserting $p$ is not considered, since more reliable sources have an opinion on matter $p$. The second graph, $p/\langle p\|\neg p\rangle$, is equivalent to $p$: the upper source serves as an arbiter between more reliable sources in case of conflict.

![Fig. 5. Non-equivalent graphs.](image)

### 4.2. Properties

To express concisely these properties, we assume to use a language $\mathcal{L}_G(\Sigma)$ that contains graphs, with the semantics given by Definition 6, but also classical connectives, with their usual semantics.

**Notations.** $D$ and $E$ will denote formulae of $\mathcal{L}_G(\Sigma)$, including formulae of the original language $\mathcal{L}$ or framed graphs. $G$, $G_1$, and $G_2$ will denote graphs.

First note that our semantics allows the replacement of equivalent formulae:

**Theorem 4.** If $D \models E$, then $G[s := E] \models G[s := D]$.

Our next theorem allows us to add or remove a frame around a single formula, or said otherwise, the graph containing a single source with formula $\phi$ has the same models as $\phi$:

**Theorem 5.** $\overline{D} \models D$.

Note that Theorem 5 above would not be valid in [49], where $\overline{\top} \models \top$.

Our study of but is subsumed by the study of graphs, since but is just a binary graph:

**Theorem 6.** $e \models D$ but $E$ iff $e \models D/E$. 
As expected, if a graph has a minimum (most reliable) source, then the models of the graph satisfy the formula of that source:

**Theorem 7.** \( G/E \models E \).

In a connected institution, Theorem 7 can be further generalized to an arbitrary subgraph instead of a single source, stating that by removing the upper part \( (G_2) \) of a graph, we add models:

**Theorem 8.** In a connected institution, \( G_2/ G_1 \models G_1 \).

When we remove a frame in the bottom of a graph, we obtain a weaker formula:

**Theorem 9.** \( G_1/|G_2| \models G_1/ G_2 \).

If a graph is satisfiable, all its formulae are satisfiable as well:

**Theorem 10.** If \( G \) is satisfiable, each \( G_x \) is satisfiable.

If it is possible to satisfy all formulae of a graph at the same time, then the graph reduces to a conjunction:

**Theorem 11.** If \( \land_{x \in S} G_x \) is satisfiable, then \( G \models \land_{x \in S} G_x \).

We can add consequences as exceptions without changing the meaning of a graph, provided the institution is smooth:

**Theorem 12.** In a smooth institution, if \( G \models E \), then \( G/E \models G \).

Disjunction semi-distributes over any graph:

**Theorem 13.** \( \models (G[s := (C \lor D)] \lor [s := (C \lor D)] \models G[s := C] \lor G[s := D]) \).

"Semi" means that equivalence is replaced by entailment; this reflects the fact that we are allowed to choose the closest models of \( C \lor D \).

The next rule can be paraphrased as follows: assume that we can satisfy the left-hand side, namely satisfy \( E_1 \) and be as close as possible to \( G_1 \) within \( E_1 \), and symmetrically. Then we satisfy \( E_1 \land E_2 \), and we are as close as possible to both \( G_1 \) and \( G_2 \).

**Theorem 14.** \( \models (G_1/E_1) \land (G_2/E_2) \models ((G_1 \land G_2)/(E_1 \land E_2)) \).

**Theorem 15** [26]. If a graph \( H \) has more edges than \( G \), then \( H \models G \).
Derived rules for but are established as corollaries in Appendix B.

4.3. Application: belief revision

The setting used in [23] is, we believe, too abstract to deal satisfactorily with this problem, since justifications of belief should be taken into account to model human reasoning.

Example 4.2. A classical story, reported in [27], shows this: Assume that I arrive, hungry, very late in a little town where I know only two McDo's p and q may be open at this hour. While approaching the town, I meet a guy with a hamburger, so that I form the belief \( p \lor q \). Coming closer, I see the lights of p, so that I now believe \( p \lor q \) but p, that is p by (W2). When I try to push the door of p, I notice that the snackbar is locked and that there is nobody inside. According to Gärdenfors' theory, I now believe p but \( \neg p \), which is \( \neg p \) in the default institution PL. A solution of this riddle (but certainly not of the general problem of belief revision) is to use our reliability graphs: In the end of the story, our belief state is then the graph \( \frac{|(p \lor q)|/p/\neg p|}{\text{PL}} \), which evaluates to \( \neg p \land q \) in PL, as expected. (end of 4.2)

Example 4.3. Another puzzle is due to [53]. Imagine a man who always wears his hat when it is raining. Now you are told that it is raining, so that you also believe that he has his hat on. Then, looking through the window, you discover that it is not raining. When modelling this story, we obtain that \((\neg \text{rain} \Rightarrow \text{hat}) \text{ but } \neg \text{rain}\) but \(\neg \text{rain}\), which evaluates to \(\neg \text{rain} \land \text{hat}\) in PL: so you still believe that the man has his hat on, just because you believed it a few seconds before. The cause of the problem is clear: we have not recorded the justification for believing hat, so that the effect persists even when the justification has disappeared. Using reliability graphs, we obtain \(\neg \text{rain} \Rightarrow \text{hat})/\text{rain}/\neg \text{rain}\), which evaluates to \(\neg \text{rain}\), as expected. Note that to represent faithfully the story, we should not insert the new data according to its temporal ordering but according to its reliability, that is \(\neg \text{rain} \Rightarrow \text{hat})||\text{rain}/\neg \text{rain}\). For this simple story, the result is the same.

(end of 4.3)

5. Related work

Although the present study was developed to fulfill our needs in structuring specifications, it is related to many fields of computer science and logic. The relation between these fields has been studied in [4, 5, 20, 25, 39, 43].

5.1. Nonmonotonic logics

5.1.1. Default logic

Default logic [47] seems an adequate device to model exceptions, since the inference rules of this logic include some antecedents, a consequent—as usual—but
also **justifications** that must be consistent before the inference rule may be applied. This fits with an (often unsound) intuitive meaning of \( D \) **but** \( E \), “accept the consequence of \( D \) as long as they are consistent with \( E \”).

This logic has to be used with some care:

1. The justifications have to be carefully crafted to allow the right level of granularity. To take a trivial example, a default like \( \forall x; p(x) \) can be translated to the normal defaults:
   - \( Mp(x)/p(x) \). In this case \( p(t) \) will be inferred when consistent, for any term \( t \). We will never be able to infer \( \forall x; p(x) \), even if no exceptions are present.
   - \( M\forall x; p(x)/\forall x; p(x) \). In this case we cannot conclude \( p(t) \), except when no exceptions are present.

These slightly paradoxical properties come from the fact that default logic is defined from a syntactical notion of provability.

2. The default theory (even though a normal default theory) may admit incompatible extensions. We prefer a skeptical approach, where a unique meaning is given to each specification.

3. Some intuitively valid conclusions may be missing. (See Example 5.1 below.)

4. Due to nonmonotonicity missing consequences may give rise to intuitively invalid consequences. (See Example 5.2 below.)

5. A finer translation would require to list explicitly the exceptions in the general rule [19], defeating our initial purpose of concision and readability.

**Example 5.1** (from [16]). We formalize *Mary and Peter tend to avoid each other at parties*, by introducing a predicate \( p \) for *is at the party*. We divide the assertion above into two defaults:

\[
P(Mary) : M\neg p(Peter)/\neg p(Peter).
\]

\[
P(Peter) : M\neg p(Mary)/\neg p(Mary).
\]

Assume we know that *Peter or Mary is at the party*, that is \( p(Peter) \lor p(Mary) \), then we can conclude nothing more and, specially, we are unable to conclude the expected \( \neg (p(Mary) \land p(Peter)) \).

(end of 5.1)

**Example 5.2.** A popular example is found in [42]: *normally non-broken things are usable*

\[
M(usable(x) \land \neg broken(x))/usable(x).
\]

*You have met a friend with a broken arm, but you don’t remember which one:*

\[
\neg broken(leftarm) \lor broken(rightarm).
\]

Since we can assert no formula of the form \( \neg broken(x) \), the default can be applied in every case and we conclude both \( usable(leftarm) \) and \( usable(rightarm) \).

(end of 5.2)
5.1.2. Preferential logics

Shoham, in [51], proposes a nonmonotonic logic parameterized by a preference relation: a conclusion is (nonmonotonically) valid if it holds in all most preferred models of the antecedents. This allows him to model known systems such as subimplication [7] and circumscription [38] (presented below). This proposal is studied in [32], which gives complete deduction systems for five families of preferential logics.

This proposal is clearly connected with our work, since for a given default $D$, the models of $D \text{ but } E$ can be considered as preferred models of $E$. In other words, in a given default institution, each default $D$ defines a preferential logic.

5.1.3. Logic of theory change

The logic of theory change [23] was originally developed to model the human process of changing mind, more formally called “belief revision”. In [23], any mental state is modelled by a theory of some given logic $(\mathcal{L}, \models)$ containing classical propositional logic. On top of this, Gärdenfors postulates a function $*$ taking a mental state and a sentence, and returning the new mental state obtained by changing mind so as to believe the sentence. This setting, we believe, is too abstract to deal satisfactorily with the problem of human belief revision, since justifications of belief should be taken into account to model human reasoning—reliability graphs are better suited, see Example 4.2.

The following postulates are considered by Gärdenfors [23] as minimal rationality requirements for belief revision:

$$K^*1. \ K^* \phi \text{ is a theory.}$$

$$K^*2. \ \phi \in K^* \phi.$$ 

$$K^*3. \ K^* \phi \subseteq Cl(K \land \phi).$$

$$K^*4. \text{ If } \neg \phi \notin K, \text{ then } Cl(K \land \phi) \subseteq K^* \phi.$$ 

$$K^*5. \ K^* \phi = \mathcal{L} \text{ iff } \models \neg \phi.$$ 

$$K^*6. \text{ If } \models \phi \leftrightarrow \psi, \text{ then } K^* \phi = K^* \psi.$$ 

$$K^*7. \ K^* (\phi \land \psi) \subseteq Cl((K^* \phi) \land \psi).$$

$$K^*8. \text{ If } \neg \psi \notin (K^* \phi), \text{ then } Cl((K^* \phi) \land \psi) \subseteq K^* (\phi \land \psi).$$

As our approach is based on a possible world semantics, these postulates do not directly apply; nevertheless we can translate them, be replacing

- $K^* \phi$ by $K \text{ but } \phi$, or equivalently, $|K/\phi|$;
- $\phi \in K$ by $K \models \phi$;
- $K_1 \subseteq K_2$ by $K_2 \models K_1$. 


Some of these postulates become theorems in our approach:

- **K** need not be valid: the language \( \mathcal{L} \) might be unable to express some expressions containing \( \text{buts} \).
- **K** results from Theorem 7.
- **K** results from Theorem 11.
- **K** also.
- **K** need be valid. There are three possible reasons for this:
  - No model of \( K \) is connected to a model of \( \phi \). This case can be eliminated by requiring *connectedness*.
  - There is an infinite regression of morphisms from \( \phi \) to \( K \), so that none is minimal. This case can be eliminated by requiring *smoothness*.
  - \( K \) is unsatisfiable.
- **K** is valid, since \( \models \phi \Leftrightarrow \psi \) yields \( \phi \models \psi \).
- **K** results from Theorem 14.
- **K** is not valid, even in a simple and intuitive default institution like \( PL \).

In summary, the postulates of [23] seem somewhat too strong. The literature contains many proposals for relaxing them (see [21]).

The following differences are important however:

- Gärdenfors [23] operates on theories: One negative consequence is that no counterfactual logic can be built by the Ramsey test on top of revision. We operate more classically on possible worlds, so that this problem does not arise.
- Gärdenfors [23] assumes that the criteria for revision depend only, but also arbitrarily, on the knowledge under revision. In our setting, the criteria for revision depend on the underlying models, and have thus to behave more regularly. The axiom Or1 reflects this regularity. On the other hand, this regularity might be exaggerated for modelling human belief revision (even the weaker assumptions of [23] are too restrictive (see Example 4.2)). For this purpose, we suggest to use the more flexible reliability graphs (Section 4).
- Gärdenfors [23] assumes that revision never introduces inconsistency (axiom **K**). Although desirable, this assumption is rarely satisfied by example institutions; among others, circumscription (or FOL==) does not satisfy it.

### 5.1.4. Circumscription

Informally, circumscription is a syntactic expression that should express that the extension of predicates is minimal.

**Predicate circumscription**

The earliest variant of circumscription is introduced in [37]. If \( T \) is a first-order formula, its *predicate circumscription* is defined as the infinite set of formulae \( \text{Circum}_T(T, P) \) containing \( T \) and all instances of

\[
T[P'] \land \forall x(P'(x) \Rightarrow P(x)) \Rightarrow \forall x(P(x) \Rightarrow P'(x))
\]
where the predicates of \( P' \) are replaced by arbitrary first-order formulae. Deduction from circumscription (either this variant or most of the following) is not trivial, since it involves guessing “good” formulae to replace \( P' \). This variant was abandoned because:

- all predicates of the signature have to be minimized, hampering thus the use of defined predicates (for instance, defining \( Q \equiv \neg P \) will prevent any minimization of \( P \) nor \( Q \));
- using a first-order scheme leads to incompleteness (see the example of [33], reproduced in [41]).

Variable circumscription
To allow some predicates to vary while others are being minimized, a simple modification is sufficient: \( \text{Circum}_1(T, P_1, Z) \) (where \( P_1 \subseteq P \) and \( Z \subseteq O \cup P \)) is defined as \( T \) and all replacements of \( P_i \) and \( Z' \) in

\[
T[P_i/Z'] \wedge \forall x(P_i(x) \Rightarrow P_1(x)) \Rightarrow \forall x(P_i(x) \Rightarrow P_i'(x)).
\]

Second-order circumscription
This definition can be made more powerful by using second-order quantification that ranges not over first-order formulae but over sets: \( \text{Circum}_2(T, P_1, Z) \) is

\[
T \wedge \forall P_i, Z'.(T[P'_i, Z'] \wedge \forall x(P_i'(x) \Rightarrow P_i(x))) \Rightarrow \forall x(P_i(x) \Rightarrow P_i'(x)).
\]

When \( Z = P \setminus P_1 \), the models of this formula are exactly the models that are minimal for the preference ordering on models: \( A \leq B \) if

- \( s_A = s_B \), where \( s \) is the unique sort;
- \( f_A = f_B \) for all \( f \in F \);
- \( p_A \leq p_B \) for all \( p \in P_1 \).

This is simply the definition of \( D \text{ but } T \) when \( D = \bigwedge_{p \in P_1} \forall x. \neg p(x) \) in the default institution \( \text{FOL}^- \) (see Section 3.1.2).

Formula circumscription
Formula circumscription [38] allows the minimization of a given formula \( E \). Formally, it is defined as follows: \( \text{Circum}_3(T, E, P_1, Z) \) is

\[
T \wedge \forall P'.(T[P', Z'] \wedge \forall x(E(P', x) \Rightarrow E(P_1, x))) \Rightarrow \forall x(E(P_1, x) \Rightarrow E(P', x)).
\]

The intent may seem similar to the intent of \( D \text{ but } T \), with \( D = \neg E \). These approaches differ in their treatment of open variables: the open variables of \( E \) determine the granularity of the minimization, since we minimize the instances of \( E \). On the other hand, in \( D \text{ but } T \), \( D \) can be a closed formula, since the granularity is fixed by the default institution. We can still model formula circumscription through an indirect translation. As indicated in [38], the use of an arbitrary expression \( E \) above can be
eliminated by introducing a fresh predicate $q$ and circumscribing $q$ in $I \land \forall x.q(x) \Leftrightarrow E(P,x)$ with $P \cup \{q\}$ varying.

**Prioritized circumscription**

Prioritized circumscription [35] is an extension that can be modelled by an expression $D_1/\cdots/D_n/E$, with $D_i = \bigwedge_{P \in P_i} \forall x.p(x)$.

**Comparison**

We have seen that second-order circumscription (with no functions varying) can be modelled by $D$. The reverse is also true: $D$ can be modelled by using four copies of the predicate $P$, say $P_d$, $P_e$, $P_{de}$, $P_{de}$, where the first two represent the extension of the predicates in the models $d$ and $e$, and the last two the differences between these two. We minimize thus the last two in

$$D[P_d/P] \land E[P_e/P] \land P_{de} \Leftrightarrow (P_e \land \neg P_d) \land P_{de} \Leftrightarrow (P_d \land \neg P_e).$$

Our definition often avoids coding tricks of circumscription, such as abnormality predicates. It also allows one to consider default institutions, such as SFOL, that handle exceptions on equality better than FOL $\models$. For instance, the hypothesis of uniqueness of names, often needed in the practical applications of circumscription, can be expressed (see Section 5.4.1).

5.2. Artificial intelligence

5.2.1. Inheritance networks with exceptions

The study of inheritance with exceptions for “semantic networks” is mainly concerned with the precedence problem, often analysed in graph-theoretic terms [8, 9, 29, 54]. Our approach, based on logic, has a richer language (that includes first-order logic) but automated reasoning is less efficient.

5.3. Logic programming

Many recent contributions to this field try to give a decent meaning to negation in logic programs.

5.3.1. Closed world assumption

This assumption, implicit in the deduction mechanism of Prolog, is often used to model nonmonotonic reasoning [46]. This interpretation of a theory $E$ can be modelled by $D \text{ but } E$, where $D$ contains $\neg p(x)$ for all $p \in P$. This expression gives the same result as the closed world assumption on Horn theories, but gives also consistent results for disjunctive theories.

5.3.2. Perfect models

The closed world interpretation is not always intuitive and convenient. A better—even called perfect—class of models is proposed in [43] for ground stratified programs. When this stratification is generated by a finite stratification, i.e. by $m$
finite sets of literals (strata) \( l_i \), the perfect models are the models of \( \neg l_1 \cdots \neg l_n / E \) in SFOL. The general case can only be treated by infinite expressions in our formalism.

5.4. Algebraic theory

5.4.1. Initial model

The case of equations (no other predicates, and no logical connectives) has been studied in [17], where they prefer the initial model, which has the properties of "no junk" (surjective) and "no confusion".

No confusion

This last property can be modelled in SFOL by

\[ D \text{ but } (\text{Equality}(\Sigma E) \land E), \]

where \( D \) contains \( \bigwedge_{s \in S} \forall x, y : s; x \neq y \), and \( E \) is the equational presentation. Note that this obviously requires that the equality is not built into the institution, for otherwise \( D \) would be plainly inconsistent.

No junk

The property of "no junk" is formally represented by the constraint of term generation, which is equivalent to a second-order induction principle [36].

To model this principle in our setting, we use an auxiliary predicate \( r \) (for "reachable") that we specify as follows:

\[
\left[ \forall x. \neg r(x) \text{ but } \bigwedge_{f \in O} \forall y \bigwedge_{i} r(y_i) \Rightarrow r(f(y)) \right] \\
\land \forall x. r(x).
\]

5.5. Object-oriented programming languages

The philosophical ideas on which our approach is based—structuring by use of inheritance and exceptions—are also used in object-oriented programming languages using multiple inheritance with exceptions. There are important differences, however:

- For efficiency, the combination of methods is programmed by the user, while this combination is based on semantics in our approach.
- For the same reason, the search for a method often occurs in some sequential order, while our approach takes a more skeptical view.

6. Conclusion

This article proposes a model-theoretic definition for combining defaults and exceptions into reliability graphs, based on the idea of closeness present in the
intuitive notion of exception. Our definition is parameterized by the logic used and this closeness between models that we call a default institution. One of these default institutions, SFOL, deals more satisfactorily with equality than circumscription; it is also amenable to a complete characterization in second-order logic, and to automated deduction (these topics are treated in [50]). Defaults institutions can also be used to define conditionals (in the style of [34]), update [30], and forgetting [28] operators.

Deduction rules for reliability graphs, valid in any default institution, are presented; however, they do not form a complete proof system. Our proposal has been designed, through its model-theoretic nature, to blend elegantly with the classical structuring concepts proposed by the algebraic school, but this blend has not been presented in the present paper. Finally, the practical usefulness of our proposal, and the methodological aspects of its use, have yet to be assessed through realistic experiments.

Appendix A. Definitions

A.1. Default institutions

Our definition is parameterized by a default institution, which is given by:

- a category \( \text{Sign} \) of signatures;
- a functor \( \text{Sen}: \text{Sign} \to \text{Sets} \), giving languages \( \mathcal{L}(\Sigma) \) linked by translations \( \text{Tr}_i \);
- a contravariant functor \( \text{Int}: \text{Sign} \to \text{Cat}^{\text{op}} \), giving interpretations \( \text{Int}(\Sigma) \) and their morphisms \( \text{Mor}(\Sigma) \), linked by forgetful functors denoted \( | \); the class of interpretations should not be empty;
- a family of satisfaction relations \( \models_X \) between the interpretations of \( \Sigma \) and its formulae;
- a functor \( \text{Comp}: \text{Sign} \to \text{Cat}^{\text{op}} \), such that for each \( \Sigma \):
  - the objects of \( \text{Comp} \) are the morphisms of \( \text{Int} \);
  - \( \text{Comp} \) is a preorder, that is, there is at most one morphism of \( \text{Comp} \) between two objects of \( \text{Comp} \);
  - the identities of \( \text{Mor} \) are initial (minima) in \( \text{Comp} \);
  - the morphisms of \( \text{Mor} \) that are minima in \( \text{Comp} \) are called agreements; they must form a subcategory \( \text{Int}_0 \);
  - \( \text{Int}_0 \) is weakly abstract: \( \exists h: M \to N \in \text{Int}_0 \Rightarrow \text{Cl}(M) \subseteq \text{Cl}(N) \);  
  - 0-symmetry: each agreement \( h: M \to N \) has a reverse agreement \( h^R: N \to M \) such that \( (h^R)^R = h \);
  - 0-equivalence: for any morphism \( h: B \to C \) and agreements \( a: A \to B \) and \( c: C \to D \), \( a = h = c \).

Note: Unlike [10], we do not require the satisfaction condition nor the soundness condition (which is ensured by construction).

When there is a morphism \( h \) such that \( \forall h' \in \text{Min}((h: M \to N)), h \equiv h' \), we define \( d(M, N) \) as \( h \). When \( d(M, N) = 0 \), \( M \) and \( N \) agree.
A.2. Graph operations

The graph $G'' = G/ G'$ is defined by:

1. $S'' = S \cup S'$.
2. $s' <_s s$, for any $s' \in S'$ and $s \in S$;
   - consequently, $s <_s s'$ is false, for any $s' \in S'$ and $s \in S$;
   - $s_1 <_s s_2$ if $s_1 <_s s_2$, for any $s_1, s_2 \in S$;
   - $s'_1 <_s s'_2$ if $s'_1 <_s s'_2$, for any $s'_1, s'_2 \in S'$.
3. $\Phi''(s) = \Phi(s)$ if $s \in S$;
   - $\Phi''(s') = \Phi'(s')$ if $s' \in S'$.

The graph $G'' = G \parallel G'$ is defined by:

1. $S'' = S \cup S'$.
2. $s' <_s s$ and $s <_s s'$ are false for any $s' \in S'$ and $s \in S$;
   - $s_1 <_s s_2$ if $s_1 <_s s_2$, for any $s_1, s_2 \in S$;
   - $s'_1 <_s s'_2$ if $s'_1 <_s s'_2$, for any $s'_1, s'_2 \in S'$.
3. $\Phi''(s) = \Phi(s)$ if $s \in S$;
   - $\Phi''(s') = \Phi'(s')$ if $s' \in S'$.

The graph $G'' = G[s := G']$ is defined by:

1. $S'' = (S \setminus \{s\}) \cup S'$.
2. $s'_1 <_s s_2$, for $s'_1 \in S'$ and $s_2 \in S$, holds iff $s <_s s_2$;
   - $s_1 <_s s_2$, for $s_1 \in S$ and $s'_2 \in S'$, holds iff $s_1 <_s s$;
   - $s_1 <_s s_2$ if $s_1 <_s s_2$, for any $s_1, s_2 \in S$;
   - $s'_1 <_s s'_2$ if $s'_1 <_s s'_2$, for any $s'_1, s'_2 \in S'$.
3. $\Phi''(s_1) = \Phi(s_1)$ if $s_1 \in S \setminus \{s\}$;
   - $\Phi''(s') = \Phi'(s')$ if $s' \in S'$.

A.3. Graph logic

The syntax of $\mathcal{L}_G(\Sigma)$ is given by:

$$\phi ::= o | \phi_1 \land \phi_2 | \top | \neg \phi_2 | G$$

where:

- $o \in \mathcal{L}(\Sigma)$ (a formula of the original language);
- $\phi_1, \phi_2 \in \mathcal{L}_G(\Sigma)$ (formulae of the extended language);
- $G$ is a reliability graph, i.e. a set of partially ordered sources labelled by formulae of $\mathcal{L}_G(\Sigma)$.

Its semantics is given by:

- $M \models o$ is given by the semantics of the original language;
- $M \models \phi \land \psi$ iff $M \models \phi$ and $M \models \psi$;
- $M \models \top$;
- $M \models \neg \phi$ iff $M \models \phi$ is false;
- $M \models |G|$ iff $\exists h \in \text{Min}(G) ; \forall s \in S ; \text{dom}(h_s) = e$. 
A.4. Properties of defaults institutions

A default institution is:

- weakly symmetric iff each morphism $h$ has a reverse $h^R$ such that:
  - if $h : A \to B$, then $h^R : B \to A$,
  - $(h^R)^R = h$,
  - if $h \leq h'$, then $h^R \leq h'^R$;
- connected iff any couple of interpretations is linked by a morphism;
- smooth iff for any graph $G$, for any $h \in Mor(G)$, there is a $h' \leq_G h$ with $h' \in Min(G)$.

Appendix B. Proofs

Theorem 1. Correspondences form a category for relational composition.

Proof. Since relations form a category, it remains to prove:

(1) A composition of correspondences is a correspondence. Let $\sim''$ be the composition of $\sim$ and $\sim'$:
(a) $\sim''$ is compatible with operators:
$$a_i \sim'' c_i$$
$$= \exists b_i \in s_{B}, a_i \sim b_i \land b_i \sim' c_i$$
$$\Rightarrow f(\ldots a_i \ldots) \sim f(\ldots b_i \ldots) \land f(\ldots b_i \ldots) \sim' f(\ldots c_i \ldots)$$
$$\Rightarrow f(\ldots a_i \ldots) \sim'' f(\ldots c_i \ldots)$$
(b) $\sim''$ is total:
$$\forall a \in s_A, \exists b \in s_B, a \sim b \land \forall b \exists c, b \sim' c$$
$$\Rightarrow \forall a \in s_A, \exists c \in s_C, a \sim'' c$$
(c) $\sim''$ is surjective by a symmetrical argument.

(2) For any algebra $A$, the identity is a correspondence between $A$ and $A$. □

Lemma B.1. If the signature is sensible, the correspondences between two algebras form a complete upper lattice for componentwise inclusion.

Proof. Given a non-empty set of correspondences $\sim_i$, we have to build the smallest correspondence $\sim$ that contains each $\sim_i$. $\bigcup_i \sim_i$ is surjective and total, but not necessarily compatible. Adding the consequences of compatibility is a monotonic function on the complete lattice of the family of relations; it has thus a unique minimal fixpoint, which is the result we look for. The top of this upper lattice is the family of full relations between carriers, i.e. $\sim = A \times B$. This is a correspondence since we consider only sensible signatures, so that no carrier is empty. □

Lemma B.2. Surjective homomorphisms [17] are functional correspondences (and the other way around).
Proof. Recall that a homomorphism is a family of total functions from $s_A$ to $s_B$ that is compatible with operators, and that a (family of) relations is functional if for any element $a$ of $s_A$, there is at most one related element of $s_B$ (this element is denoted $\sim(a)$). For correspondences, it is known that $\sim(a)$ indeed exists. The definitions are thus just identical. $\square$

Theorem 2. The correspondences between two surjective algebras form a complete lattice.

Proof. The upper part is proved in Lemma B.1. We just have to prove the lower part, i.e. to construct $\sim$, the largest correspondence contained in $\sim_i$. $\bigcap_i \sim_i$ is compatible with operators, but not total and surjective for non-surjective algebras. Here however, each data element of $s_A$ is represented by a term $t \in T_{sA}$. $e^R_A$, the relational inverse of $e_A$, is thus a correspondence. $e^R_A$, $e_B$, which we call the ground correspondence, is a correspondence by Theorem 1. We can check by induction that each compatible relation contains the ground correspondence, so that the bottom of the lattice of correspondences is indeed $e^R_A$, $e_B$. $\square$

Lemma B.3. Stronger correspondences yield stronger disagreements: $\sim_1 \subseteq \sim_2$ $\Rightarrow \sim_1 \preceq \sim_2$.

Lemma B.4. In SFOL, morphisms are agreements iff they are compatible with predicates.

Proof. By definition, a morphism $\sim : A \rightarrow B$ is an agreement if $\sim \leq id$, for some identity $id$. Using the definition of $\leq$ for SFOL $\forall a, b : a \sim b \Rightarrow p_A(a) \Leftrightarrow p_B(b)$, as wanted. $\square$

Lemma B.5. SFOL-agreements form a category whose objects are SFOL-interpretations.

Proof. Using Theorem 1, we just have to show:

1. The identity is an agreement between $A$ and $A$, easy.
2. The composition of agreements is an agreement. From $a \sim b \Rightarrow p_A(a) \Leftrightarrow p_B(b)$ and $b \sim c \Rightarrow p_B(b) \Leftrightarrow p_C(c)$, transitivity of equivalence shows that $a \sim c \Rightarrow p_A(a) \Leftrightarrow p_C(c)$. $\square$

Lemma B.6. The reverse of a SFOL-agreement is a SFOL-agreement.

Proof. By Lemma B.4. $\square$

Lemma B.7. Any correspondence weaker than an agreement is an agreement.

Proof. By Lemma B.3. $\square$
Theorem 3. If two SFOL-interpretations agree, they satisfy the same first-order formulae.

Proof. Two valuations are related by a correspondence ~ if all their elements correspond: \( V \sim V' \iff \forall x \in X, V(x) \sim V'(x) \). Given a valuation \( V \) and a correspondence ~, it is always possible to find a corresponding valuation \( V' \) such that \( V \sim V' \). In this case, \( V_A(t) \sim V'_A(t) \). Furthermore, if \( V_1 \approx V \), then the corresponding \( V'_1 \approx V' \). Let us prove \( A \models_V f \iff B \models_{V'} f \) when \( V \sim V' \) (where ~ is the given agreement) by induction:

\[
\begin{align*}
B \models V p(t_1, \ldots, t_n) & \iff p_B(V_B(t_1), \ldots, V_B(t_n)) & \text{Definition} \\
& \iff p_A(V_A(t_1), \ldots, V_A(t_n)) & \text{Theorem 4} \\
& \iff A \models_V p(t_1, \ldots, t_n); & \text{Definition}
\end{align*}
\]

\[
\begin{align*}
B \models V \forall v, f & \iff \forall V_1 \approx V', B \models_{V'} f & \text{Definition} \\
& \iff \forall V_1 \approx V, A \models_{V_1} f & \text{Induction hypothesis} \\
& \iff A \models_V \forall v, f; & \text{Definition}
\end{align*}
\]

\[
\begin{align*}
B \models V f_1 \land f_2 & \iff B \models_V f_1 \text{ and } V \models_V f_2 & \text{Definition} \\
& \iff A \models f_1 \text{ and } A \models_V f_2 & \text{Induction hypothesis} \\
& \iff A \models_V f_1 \land f_2; & \text{Definition}
\end{align*}
\]

\[
\begin{align*}
B \models V \neg f & \iff \neg B \models_V f & \text{Definition} \\
& \iff \neg A \models_V f & \text{Induction hypothesis} \\
& \iff A \models_V \neg f. & \text{Definition}
\end{align*}
\]

Lemma B.8. In SFOL, "\( \leq \)" is a preorder.

Proof.

Reflexivity: \( \sim : A \rightarrow B \) is less than itself using the identity as the double correspondence needed to make the comparison.

Transitivity: For \( \sim : A \rightarrow B, \sim' : A' \rightarrow B' \), and \( \sim'' : A'' \rightarrow B'' \), if \( \sim \leq \sim' \) (using \( = \)) and \( \sim' \leq \sim'' \) (using \( =' \)), then we show that \( \sim \leq \sim'' \) using the relational composition \( \sim ; \sim' \).

Lemma B.9. 0-equivalence: For any SFOL-correspondence \( \sim : B \rightarrow C \) and SFOL-agreements \( \sim_1 : A \rightarrow B \) and \( \sim_2 : C \rightarrow D, \sim_1 \approx \sim_2 \approx \sim_1 \).

Proof. We only show \( \sim_1 \approx \sim_2 \), as the other part is symmetrical.

(\( \sim_1 \approx \sim_2 \)) We define the double correspondence by \( (b, c) \approx (a, c') \iff a \sim_1 b \land c = c' \). If \( p_B(b), a \sim_1 b, \neg p_C(c) \), then as \( a \sim_1 b \) we have \( p_A(a) \iff p_B(b) \) by Lemma B.4, so that \( p_A(a), a \sim_1 c \sim_2 c, p_C(c) \). (The case \( \neg p_B(b), b \approx c, p_C(c) \) is symmetrical.)
We define the double correspondence by \((a, c) = (b, c') \iff a \sim_1 b, b \sim c, c = c'.\) If we have a conflict \(p_A(a), a \sim_1 \;\sim c, \neg p_C(c),\) then all corresponding pairs \((b, c)\) are such that \(a \sim_1 b,\) and thus \(p_A(a) \iff p_A(b),\) from which we conclude \(p_B(b), b \sim c, \neg p_C(c).\) (The case \(\neg p_A(a), a \sim_1 \;\sim c, p_C(c)\) is symmetrical.)

**Lemma B.10.** If \(h \in \text{Min}(T), h \in S,\) and \(S \subseteq T,\) then \(h \in \text{Min}(S).\)

**Proof.**

\[
\begin{align*}
\text{Definition of Min} & \quad \Rightarrow \quad h \in T \text{ and } \forall h' \in T, h' \prec h \\
\Rightarrow \quad h \in T \text{ and } \forall h' \in S, h' \not\prec h \\
\Rightarrow \quad h \in S \text{ and } \forall h' \in S, h' \not\prec h \\
\quad \quad h \in S \\
\quad \quad \Rightarrow \quad h \in \text{Min}(S). \quad \square
\end{align*}
\]

**Lemma B.11.** If \(S \cap \text{Int}_0\) is not empty, then \(\text{Min}(S) = S \cap \text{Int}_0.\)

**Proof.** Let \(h_0\) be some agreement in \(S; h_0 \equiv 0.\)

\[
\begin{align*}
(\Rightarrow) & \quad h \in \text{Min}(S) \\
& \quad \Leftrightarrow h \in S, \forall h' \in S, h' \not\prec h \\
& \quad \quad \Rightarrow h \in S, h \equiv h_0 \\
& \quad \quad \Rightarrow h \in S, h = 0 \\
& \quad \quad \Rightarrow h \in S \cap \text{Int}_0. \quad \square
\end{align*}
\]

\[
\begin{align*}
\quad (\Leftarrow) & \quad h \in S \cap \text{Int}_0 \\
& \quad \Leftrightarrow h \in S, h \equiv 0 \\
& \quad \quad \Rightarrow h \in S, \forall h', h' \succ h \\
& \quad \quad \Rightarrow h \in S, \forall h', h' \not\prec h \\
& \quad \quad \Rightarrow h \in \text{Min}(S). \quad \square
\end{align*}
\]

**Lemma B.12.** \(\text{Min}(S_1 \cup S_2) \subseteq \text{Min}(S_1) \cup \text{Min}(S_2).\)

**Proof.**

\[
\begin{align*}
\quad h \in \text{Min}(S_1 \cup S_2) \\
& \quad \Leftrightarrow h \in S_1 \cup S_2, \forall h' \in S_1 \cup S_2, h' \not\prec h \\
\quad \quad \text{Definition of Min} \quad \square
\end{align*}
\]

If \(h \in S_1, h \in \text{Min}(S_1)\) and if \(h \in S_2, h \in \text{Min}(S_2).\) Thus \(h \in \text{Min}(S_1) \cup \text{Min}(S_2). \quad \square
\]

**Lemma B.13.** If \(h \prec g, h \in S, g \in \text{Min}(S),\) then \(h \in \text{Min}(S).\)

**Proof.** By contraposition:

\[
\begin{align*}
\quad h \notin \text{Min}(S) \\
& \quad \Rightarrow \exists h' \in S, h' < h \\
& \quad \text{Definition of Min} \\
& \quad \Rightarrow h' < g \\
& \quad \text{Transitivity} \\
& \quad \Rightarrow g \notin \text{Min}(S). \quad \square
\end{align*}
\]
Lemma B.14. The class of models of a class of formulae is weakly abstract.

Proof. \( A \to B \) is defined as \( \text{Cl}(A) \subseteq \text{Cl}(B) \) (\( B \) has more properties). \( A \models \Phi \iff \Phi \subseteq \text{Cl}(A) \), so that \( \Phi \subseteq \text{Cl}(B) \): \( B \) is also a model of all formulae in the given class. \( \square \)

Lemma B.15. A weakly abstract class of interpretations is closed under agreement.

Proof. As our default institutions are required to be weakly abstract. \( \square \)

Lemma B.16. “agrees” is an equivalence between interpretations.

Proof. Reflexivity and transitivity are ensured by the fact that \( \text{Int}_0 \) is a category; symmetry is obtained from \( 0 \)-symmetry. \( \square \)

Lemma B.17. \( (\forall i \in I \forall D_i) \) but \( (\forall j \in J \forall E_j) \equiv \bigvee_{(i,j) \in K} \text{Min}(D_i) \) but \( E_j \), where \( K \) contains the pairs \( (i, j) \) such that \( d(D_i, E_j) \) is minimal among all \( d(D_i, E_j) \mid i \in I, j \in J \). (All \( d(D_i, E_j) \) must be defined to apply the theorem.)

Proof. First note that if \( d(D_i, E_j) \) is defined, \( h \in \text{Min}(E_j, D_i) \) iff \( h \in \text{Mor}(E_j, D_i) \) and \( h = d(D_i, E_j) \). By definition of \( \text{but} \), it suffices to show that

\[
\text{Min}_{(i,j) \in K} \left( \bigvee_{i \in I} E_i, \bigvee_{i \in I} D_i \right) = \bigcup_{(i,j) \in K} \text{Min}(E_j, D_i).
\]

\((\subseteq)\) Let \( h \in \text{Min}_{(i,j) \in K} \left( \bigvee_{i \in I} E_i, \bigvee_{i \in I} D_i \right) \); therefore \( h \in \text{Mor}(E_j, D_i) \) for some \( i \) and \( j \), and by Lemma B.12, \( h \in \text{Min}(E_j, D_i) \), so that \( h = d(D_i, E_j) \). If \( d(D_i, E_j) \) is not minimal, then \( h \in \text{Min}_{(i,j) \in K} \left( \bigvee_{i \in I} E_j, \bigvee_{i \in I} D_i \right) \), contradicting our assumption. Thus \( h \in \bigcup_{(i,j) \in K} \text{Min}(E_j, D_i) \).

\((\supseteq)\) Let \( h \in \bigcup_{(i,j) \in K} \text{Min}(E_j, D_i) \), i.e. \( h \in \text{Min}(E_j, D_i) \) (equivalently, \( h = d(D_i, E_j) \)) for some \( (i, j) \in K \). Assume for \( h \) the negation of the thesis, i.e. there is a \( h' \in \text{Mor}(\bigvee_{i \in I} E_i, \bigvee_{i \in I} D_i) \), \( h' < h \); then \( h' \in \text{Mor}(E_j, D_i) \) for some \( i' \) and \( j' \). Consequently, \( d(D_i, E_j) \) is minimal among all \( d(D_i, E_j) \mid i \in I, j \in J \), contradicting that \( (i, j) \in K \). \( \square \)

Lemma B.18. The ordering among families is a preorder.

Proof.

Reflexivity: From the reflexivity of the morphism order.

Transitivity: Assume \( \tilde{h} \leq \tilde{h}' \leq \tilde{h}'' \), and that \( h_s \neq h''_s \) for some \( s \). Our goal is to prove that \( h_u < h''_u \) for some \( u < s \).

Case 1. \( h \leq h' \).

- \( h' \neq h''_u \) is impossible by transitivity of the morphism order.
- \( h' \neq h''_u \) implies the existence of a \( t < s \) such that \( h'_t < h''_t \). Let's take a minimal \( t \).
  - if \( h_t \leq h'_t \), by transitivity we have \( h_s < h''_s \);
  - if \( h_t \neq h'_t \), then we have \( u < t \) such that \( h_u < h'_u \). \( h'_t < h''_t \) because otherwise \( t \\) is not minimal. By transitivity we obtain \( h_u < h''_u \).
Case 2: \( h \neq h' \). Let us take again a minimal \( t \) such that \( h_t < h'_t \).

- if \( h'_t \leq h''_t \), we use transitivity;
- if \( h'_t \neq h''_t \), we have a \( u < t \) such that \( h'_u < h''_u \), and \( h_u \leq h''_u \) by minimality of \( t \).

Lemma B.19. The comparison between two families \( \tilde{f} \) and \( \tilde{f}' \) is determined by their frontier, i.e. the minimal sources where \( f \neq f' \):

1. \( \tilde{f} = \tilde{f}' \) iff \( \text{fr}(\tilde{f}, \tilde{f}') = \emptyset \);
2. else,
   - \( \tilde{f} < \tilde{f}' \) iff \( \forall s \in \text{fr}(\tilde{f}, \tilde{f}'), f_s < f'_s \);
   - \( \tilde{f} > \tilde{f}' \) iff \( \forall s \in \text{fr}(\tilde{f}, \tilde{f}'), f_s > f'_s \);
   - else, \( \tilde{f} \neq \tilde{f}' \) (\( f \) and \( f' \) are incomparable).

Proof. We first prove (1):

\[ \Rightarrow \ h = \tilde{h}' = \tilde{h} \]
\[ = \forall s \in S, h_s \neq h'_s \Rightarrow \exists t < s, h_t < h'_t, \]
\[ \quad \forall s \in S, h_s = h'_s \Rightarrow \exists t < s, h_t > h'_t. \]

Assume \textit{ad absurdum} \( h_s \neq h'_s \). We consider the cases \( h_s \neq h'_s \) and \( h_s \neq h'_s \) separately.

\[ h_s \neq h'_s \Rightarrow h_t < h'_t, \quad \text{because} \quad \tilde{h} \leq \tilde{h}' \]
\[ h_s \neq h'_s \Rightarrow h_t > h'_t, \quad \text{because} \quad \tilde{h} \geq \tilde{h}'. \]

This creates an infinite decreasing sequence in \( S \), and this is impossible by the well-foundedness of \( S \).

For \( h_s \neq h'_s \), we have a symmetrical impossibility.

\( \Leftarrow \) Obvious.

Now we prove (2a):

\[ \Rightarrow \] Let \( s \in \text{fr}(\tilde{f}, \tilde{f}') \). If \( f_s \neq f'_s \), then \( \exists t < s, f_t < f'_t \), so that \( s \) is not minimal, impossible. If \( f_s = f'_s \), then \( f_s < f'_s \), since \( f \neq f' \).

\( \Leftarrow \) To prove \( \tilde{f} \leq \tilde{f}' \) assume \( f \neq f' \). Since \( S \) is well-founded, \( \exists t \in \text{fr}(f, f'), t \leq s \).

The proof of (2b) is symmetrical, and (2c) is the only remaining case.

Lemma B.20. If \( G = G_1 / G_2 \), then \( \tilde{h}' <_G \tilde{h} \) iff \( (\tilde{h}'|_{G_1} <_{G_1} \tilde{h}|_{G_1} \text{ or } \tilde{h}'|_{G_1} =_{G_1} \tilde{h}|_{G_1} \text{ and } \tilde{h}'|_{G_2} <_{G_2} \tilde{h}|_{G_2}) \).

Proof. As \( \tilde{h}' <_{G_1} \tilde{h} \), by Lemma B.19 there is a non-empty frontier \( \text{fr} \subseteq S \) such that \( \forall s \in \text{fr}, h'_s < h_s \). As every node of \( S \) is in some \( S_i \), there is an \( S_i \) such that \( \text{fr} \cap S_i \neq \emptyset \).

The frontier cannot be partly in both graphs: if we assume that there are \( s \in \text{fr} \cap G_1 \) and \( s' \in \text{fr} \cap G_2 \), we have \( s' <_S s \) contradicting our definition of frontier as minimal sources. If the frontier is in the lower graph, we have \( \tilde{h}'|_{G_1} <_{G_1} \tilde{h}|_{G_1} \); if it is in the upper graph, we have \( \tilde{h}'|_{G_2} <_{G_2} \tilde{h}|_{G_2} \). \( \Box \)
Lemma B.21. If $G = G_1 \parallel G_2$, then $h^* <_G h$ iff $h^*\mid_{G_1} <_G h\mid_{G_2}$ for some $i \in \{1, 2\}$.

Proof. As $h^* <_G h$, by Lemma B.19 there is a non-empty frontier $fr \subseteq S$ such that $\forall s \in fr$, $h'_s < h_s$. As every node of $S$ is in some $S_i$, there is an $S_i$ such that $fr \cap S_i \neq \emptyset$. Assume there is an $s \in fr$ (the frontier for $G_i$) that does not belong to $fr$. Since $G$ is well-founded there is an $s' < s$, $s' \in fr$. But if $s' < s$ then $s' \in S_i$, so that $s \notin fr$. Thus $fr = fr \cap S_i$. □

Theorem 4. If $D \models E$, then $G[s := E] \models G[s := D]$.

Proof. Immediate, since the semantics of graphs only considers the models of the component formulae. □

Theorem 5. $\models D \models D$.

Proof.
$$
eq \iff \exists h : e \rightarrow d \in Min(\tau, D)
\iff h = 0
\iff \models D \quad \text{by closure.}$$ □

Theorem 6. $\models D$ but $E$ iff $\models D/E$

Proof.
$(\Rightarrow)$ If there is a minimal morphism $h$ for $D$ but $E$, we can extend it to a pair $f = (id, h)$, which is minimal for the lexicographic ordering.

$(\Leftarrow)$ Let us call $s_0$ the source of $E$, and $s_1$ the source of $D$. If $E$ is unsatisfiable, then there are no morphisms of codomain $e$, and therefore no models, as in $D$ but $E$. If $E$ has a model, its identity is less than any other morphism, so that a pair can only be minimal if its $s_0$ component has null distance. Therefore its domain $e$ agrees with its codomain (a model of $E$), that is $e$ is a model of $E$ by closure of $E$. □

Theorem 7. $G/E \models E$.

Proof. Let $e$ be a model of $G/E$; there is thus a minimal family $\tilde{h}$, with $h_i : e \rightarrow d_i$. Let 0 be the minimum source of $G/E$, and let $\tilde{h}'$ be as $\tilde{h}$ ($s \neq 0 \Rightarrow h'_s = h_s$), except for 0: $h'_0 = id$. $\tilde{h}' \models \tilde{h}$, since $h'_i \leq h_i$. If $\tilde{h}' < \tilde{h}$, then $h$ would not be minimal, contradicting our hypothesis, so that $\tilde{h}' \models \tilde{h}$ and thus $h'_0 = h_0$, which means that $h_0$ is an agreement, and thus $\models E$ by closure. □

Theorem 8. In a connected institution, $G_2/G_1 \models G_1$.

Proof. If one of the formulae of $G = G_2 \parallel G_1$ has no model, then $G$ has no model and the property is vacuously true. Else, if $\models G$, there is a minimal family $\tilde{h}$ of
domain $e$, thus all $h'$ of domain $e'$ are not smaller; thus the part relative to $G_1$ ($h'|_{G_1}$) is not smaller than $h|_{G_1}$. Now for any $f'$, the family of morphisms for $G_1$ is of the form $h'|_{G_1}$, because we can complete $f'$ by taking for each $s_2 \in G_2$ a morphism (whose existence is guaranteed by connectedness) from $e'$ to a model of $G_2$ (whose existence is assumed in this case of the proof).

**Theorem 9.** $G_1/|G_2| \models G_1/G_2$.

**Proof.** Let $e \models G_1/|G_2|$, which means $\exists h \in \text{Min}(G_1/|G_2|)$, $\text{dom}(h) = e$. Let $s_0$ be the minimum source, whose formula is $|G_2|$. By Theorem 7, $h_{s_0}$ is an agreement, so that $e \models G_2$, which means $\exists h_2 \in \text{Min}(G_2)$, $\text{dom}(h_2) = e$. It suffices now to prove that $h|_{G_1}/h_2 \in \text{Min}(G_1/G_2)$ to obtain the desired result, since $\text{dom}(h|_{G_1}/h_2) = e$. Assume ad absurdum that there exists $p < h|_{G_1}/h_2$, with $h' \in \text{Mor}(G_1/G_2)$. By Lemma B.20, either $h'|_{G_1} < h_2$, which would contradict $h \in \text{Min}(G_1/G_2)$; or $h|_{G_1} < h'|_{G_1}$, so that $h|_{G_1}/\text{id}_{\text{dom}(h')} < h|_{G_1}/\text{id}_e = h$, contradicting $h \in \text{Min}(G_1/|G_2|)$.

**Theorem 10.** If $G$ is satisfiable, each $G_s$ is satisfiable.

**Proof.** By contraposition, if some $G_s$ is unsatisfiable, then we are unable to construct a morphism (and thus a family of morphisms) to one of its models.

**Theorem 11.** If $\wedge_{s \in S} G_s$ is satisfiable, then $G \models \wedge_{s \in S} G_s$.

**Proof.**

$(\Rightarrow)$ If $e \models \wedge_{s \in S} G_s$, then $\tilde{id}_e$ is a minimum (and thus minimal) family showing that $e \models G$.

$(\Leftarrow)$ There is a model $e \models \wedge_{s \in S} G_s$. As the family of morphisms $\tilde{id}_e$ is minimum, every other minimal family is a family of agreements, and thus any model of $G$ satisfies each $G_s$ by closure, and thus their conjunction.

**Theorem 12.** In a smooth institution, if $G \models E$, then $G/E \models G$.

**Proof.**

$(\Rightarrow)$ $e \models G/E$

$\iff \exists h \in \text{Min}(G/E)$, $\text{dom}(h) = e$

$\Rightarrow h|_G \in \text{Mor}(G)$.

**Definition 6**

smoothness

Assume $h|_G \notin \text{Min}(G)$. Then

$\exists h' <_G h|_G$, $h' \in \text{Min}(G)$

$\Rightarrow \text{dom}(h') = e \models E$

$\Rightarrow h'/\text{id}_e < h$

$\Rightarrow h \notin \text{Min}(G/E)$. 

hypothesis
\[(\Leftrightarrow) \quad e \models G \quad \Rightarrow \quad e \models E \quad \text{hypothesis} \]
\[\Rightarrow \exists h \in \text{Min}(G), \ dom(h) = e \quad \text{Definition 6} \]
\[\Rightarrow h/\text{id}_e \in \text{Mor}(G/E). \]

Assume \(h/\text{id}_e \notin \text{Min}(G/E).\) Then
\[\exists h' < h/\text{id}_e, h' \in \text{Mor}(G/E) \]
\[\Rightarrow h'\big|_G < h \]
\[\Rightarrow h \notin \text{Min}(G). \quad \square \]

**Theorem 13.** \(|G[s := (C \vee D)]| = |G[s := C]| \vee |G[s := D]|.\)

**Proof.**
\[e \models |G[s := (C \vee D)]| \quad \Leftrightarrow \quad \exists \bar{h} : e \rightarrow d_i \in \text{Min}(G[s := (C \vee D)]) \quad \text{Definition 6} \]

Either \(d_i \models C\) and then \(\bar{h} \in \text{Min}(G[s := C])\) by Lemma B.12, or \(d_i \models D\), and thus \(h \in \text{Min}(G[s := D])\) symmetrically. \(\square\)

**Theorem 14.** \(|G_1/E_1| \land |G_2/E_2| = |(G_1||G_2)/(E_1 \land E_2)|.\)

**Proof.**
\[e \models |G_1/E_1| \land |G_2/E_2| \quad \Rightarrow \quad \exists h_1 \in \text{Min}(G_1/E_1), \ dom(h_1) = e \land \exists h_2 \in \text{Min}(G_2/E_2), \ dom(h_2) = e \quad \text{Definition 6} \]

Let \(h = (h_1|_{G_1}| h_2|_{G_2})/\text{id}_e.\) Then \(h \in \text{Mor}((G_1||G_2)/(E_1 \land E_2)).\)

Assume \textit{ad absurdum} that \(h\) is not minimal, i.e. \(h' < h.\) By Lemma B.19 there is a non-empty frontier \(fr\) such that \(\forall s \in fr, \ h' < h_s.\)

- The source of \(E_1 \land E_2\) cannot be in the frontier, since identities are minimal.
- By Lemma B.21, if the frontier intersects with \(S_1\) then \(h'\big|_{G_1} < h_1|_{G_1},\) and thus \(h'|_{G_1/E_1} < h_1,\) contradicting the minimality of \(h_1.\)
- Symmetrically for \(S_2.\) \(\square\)

**Lemma B.22.** \(|G/D| \land |G/E| \models |G/(D \land E)|.\)

**Proof.** Let \(e\) be a model of \(G/D\) and \(G/E.\) There are thus two minimal families of morphisms \(\bar{h}\) and \(\bar{h}'\) of domain \(e\) for \(G/D\) and \(G/E\) respectively. By reasoning in a similar way as in the proof of Theorem 14, we have \(h \in \text{Min}(G/(D \land E)). \quad \square\)

**Lemma B.23.** \(|G| \land E \models |G/E|.\)
On the meaning of "but"

Theorem 15. If a graph $H$ has more edges than $G$, then it has less models: $H \models G$.

Proof. First note that adding edges reduces the frontier between two families: $fr_H(f,f') \subseteq fr_G(f,f')$ but cannot void it due to well-foundedness:

$$fr_G(f,f') \neq \emptyset \Rightarrow fr_H(f,f') \neq \emptyset.$$  

Consequently, if two families are equivalent, strictly better, or strictly worse for $G$, they will still be for $H$. If two families are incomparable, they cannot become equivalent. Here, we only need that $f' \prec_G f \Rightarrow f' \prec_H f$. Thus, if a family is minimal for $H$ ($f \in \text{Min}(H)$, i.e. $\exists f' \in \text{Mor}(H), f' \prec_H f$) it is also minimal for $G$. □

Corollary 1. (S) $(\phi \text{ but } \psi) \Rightarrow \psi$.

Proof. By Theorem 7. □

Corollary 2. (W1) From $\psi \Rightarrow \phi$, infer $(\psi \text{ but } \phi) \Leftrightarrow \psi$.

Proof. If $\psi$ is satisfiable, by Theorem 11 we have $\psi \text{ but } \phi \models \psi \land \phi \models \psi$. If $\psi$ is not satisfiable, by Theorem 10 we have $\psi \text{ but } \phi \models \perp \models \psi$. □

Corollary 3. (Or1) $((\psi_1 \lor \psi_2) \text{ but } \phi) \Rightarrow (\psi_1 \text{ but } \phi) \lor (\psi_2 \text{ but } \phi)$.


Corollary 4. (K7) $(\psi \text{ but } \phi) \land \phi_2 \Rightarrow \psi \text{ but } (\phi \land \phi_2)$.


Corollary 5. (T1) $(\psi \text{ but } \top) \Leftrightarrow \psi$.

Proof. Since $\psi \Rightarrow \top$, we have by (W1) $(\psi \text{ but } \top) \Rightarrow \psi$. □

Corollary 6. (Or2) $\psi \text{ but } (\phi \lor \phi_2) \Rightarrow (\psi \text{ but } \phi) \lor (\psi \text{ but } \phi_2)$.

Proof. Using (K7) with $\phi = \alpha \lor \beta$ twice; first with $\phi_2 = \alpha$, then, with $\phi_2 = \beta$:

$(\psi \text{ but } (\alpha \lor \beta)) \land \alpha \Rightarrow (\psi \text{ but } \alpha)$,

$(\psi \text{ but } (\alpha \lor \beta)) \land \beta \Rightarrow (\psi \text{ but } \beta)$. 

Proof.
The disjunction of the antecedents is

\[ ((\psi \text{ but } (\alpha \lor \beta)) \land \alpha) \lor ((\psi \text{ but } (\alpha \lor \beta)) \land \beta), \]

which distributes into \((\psi \text{ but } (\alpha \lor \beta)) \land (\alpha \lor \beta)\). Since \((\psi \text{ but } (\alpha \lor \beta)) \Rightarrow (\alpha \lor \beta)\) by (S), we obtain \((\psi \text{ but } (\alpha \lor \beta)) \Rightarrow (\psi \text{ but } \alpha) \lor (\psi \text{ but } \beta)\), as desired. 

Corollary 7. (K3) \((\psi \land \phi) \Rightarrow (\psi \text{ but } \phi)\).

Proof. Take (K7) with \(\phi = \top\), i.e. \((\psi \text{ but } \top) \land \phi_2 \Rightarrow \psi \text{ but } (\top \land \phi_2)\). By (T1), we obtain \((\psi \land \phi_2) \Rightarrow (\psi \text{ but } \phi_2)\). □

Corollary 8. (T2) \((\top \text{ but } \psi) \Leftrightarrow \psi\).

Proof. By (K3) we have \((\top \land \psi) \Rightarrow (\top \text{ but } \psi)\); and by (S) we obtain \((\top \text{ but } \psi) \Rightarrow \psi\). □

Corollary 9. (ID) \((\psi \text{ but } \psi) \Leftrightarrow \psi\).

Proof. By (W1). □

Corollary 10. (ID2) \((\psi \text{ but } \phi) \text{ but } \phi) \Leftrightarrow (\psi \text{ but } \phi)\).

Proof. By (W1) and (S). □

Corollary 11. (W2) From \(\phi \Rightarrow \psi\), infer \((\psi \text{ but } \phi) \Leftrightarrow \phi\).

Proof. From \(\phi \Rightarrow \psi\) we deduce \(\psi \land \phi \Leftrightarrow \psi \land \phi\) and by (K3) \((\psi \land \phi) \Rightarrow (\psi \text{ but } \phi)\) so that \(\phi \Rightarrow (\psi \text{ but } \phi)\). On the other hand, (S) gives \((\psi \text{ but } \phi) \Rightarrow \phi\). □

Corollary 12. (F1) \((\psi \text{ but } \bot) \Leftrightarrow \bot\).

Proof. By (W2). □

Corollary 13. (F2) \((\bot \text{ but } \psi) \Leftrightarrow \bot\).

Proof. By (W1). □

Corollary 14. (HL) \(\psi \text{ but } \phi \Rightarrow (\psi \lor (\psi \text{ but } \phi)) \land \phi\).

Proof. Its clausal form is \(((\psi \text{ but } \phi) \Rightarrow \phi) \land ((\psi \land \phi) \Rightarrow (\psi \text{ but } \phi))\), which is proved by (S) and (K3). □

Corollary 15. (E1) \(\neg(\psi \text{ but } \phi) \Rightarrow (\psi \lor \neg \phi)\).

Proof. The contrapositive of (K3). □
Corollary 16. (11) From $\neg \psi$ infer $\neg (\psi \text{ but } \phi)$.

Proof. $\neg \psi$ is $\psi \iff \bot$; $(\psi \text{ but } \phi) \iff (\bot \text{ but } \phi)$ by Theorem 4; and $(\bot \text{ but } \phi) \iff \bot$ by (F2). □

Corollary 17. (12) From $\neg \phi$ infer $\neg (\psi \text{ but } \phi)$.

Proof. The proof is symmetrical to that of Corollary 16. □

Corollary 18. (And) $(\psi \text{ but } \phi) \land (\psi \text{ but } \phi_2) \Rightarrow \psi \text{ but } (\phi \land \phi_2)$.

Proof. From (S), we have $(\psi \text{ but } \phi_2) \Rightarrow \phi_2$. Therefore

$$(\psi \text{ but } \phi) \land (\psi \text{ but } \phi_2) \Rightarrow (\psi \text{ but } \phi) \land \phi_2,$$

and (K7) is $(\psi \text{ but } \phi) \land \phi_2 \Rightarrow (\psi \text{ but } (\phi \land \phi_2))$, giving the result by transitivity. □

Corollary 19. (CO1) From $\psi \lor \psi_2$ infer $(\psi \text{ but } \phi) \lor (\psi_2 \text{ but } \phi) \iff \phi$.

Proof. Symmetric to the first part of (CO1). □

Corollary 20. (N1) $(\psi \text{ but } \phi) \lor (\neg \psi \text{ but } \phi) \iff \phi$.

Proof. By (CO1). □

Corollary 21. (CO2) From $\phi \lor \phi_2$ infer $\psi \Rightarrow ((\psi \text{ but } \phi) \lor (\psi \text{ but } \phi_2))$.

Proof. Symmetric to the first part of (CO1). □

Corollary 22. (N2) $\psi \Rightarrow ((\psi \text{ but } \phi) \lor (\psi \text{ but } \neg \phi))$.

Proof. By (CO2). □

Corollary 23. From $(\psi \text{ but } \phi) \Rightarrow \phi_2$ and $(\psi \text{ but } (\phi \land \phi_2)) \Rightarrow \phi_3$, infer $(\psi \text{ but } \phi)$

$\Rightarrow \phi_3$.

Proof. Using (K7). □

References


On the meaning of "but"


