Propositional Logic for Ground Semigroups of Context

ROLF NOSSUM, *Agder University College, Norway,*
Rolf.Nossum@hia.no

Abstract

A propositional framework of formal reasoning is proposed, which emphasises the pattern of entering and exiting context. Contexts are modelled by an algebraic structure which reflects the order and manner in which context is entered into and exited from.

The equations of the algebra partitions context terms into equivalence classes. A formal semantics is defined, containing models that map equivalence classes of certain context terms to sets of interpretations of the formula language. The corresponding Hilbert system incorporates the algebraic equations as axioms asserted in context.

In semigroups of contexts, where combination of contexts is associative, finite ground algebraic equations correspond to contingent equivalence between certain logical formulas. Systems for sets and multisets of contexts are obtained by presenting their respective algebras as associativity plus finite ground equations. Soundness and completeness results are proved.

Some contextual reasoning systems in the literature are inherently associative, and we present those as special cases.

Keywords: Formalization of context, logic of contextual assertions, algebras of context

1 Introduction

It is commonly held that all reasoning takes place within context of some sort or another. The seminal papers of Giunchiglia et.al. [11, 10, 18] and McCarthy et.al. [16, 15], which call for a treatment of contexts by mathematical and logical methods, have spurred quite a bit of effort towards devising formal systems of reasoning that reflect this adequately.

Context tends to change in the course of reasoning, for instance if a reasoner is concerned with several independent tasks in quasi-parallel, or if she(/he/it) encounters a very general subtask that can be solved in a very large class of contexts, or a very special subtask that can only proceed within some particular and highly specific context. Thus entering and exiting contexts are fundamental operations in contextual reasoning.

In logical terms, assuming a language of contexts and a language of formulas, the pattern of entering and exiting context can be modelled by deduction rules specifying the context before and after transition, and what formulas are taken to hold in the old and the new context.

Following [12, 16, 4] we adopt the notation

$$x : \lambda \tag{1.1}$$

for asserting a formula λ in context x, and the formula syntax

$$ist(c, \lambda) \tag{1.2}$$

for truth of λ in context c. Here, c is an atomic context name, while x denotes an accumulated context composed, in general, of various atomic constituents. The syntax of the special *ist* predicate is restricted to atomic contexts in its first coordinate. Assertions of the form (1.1) are used to keep track of the surrounding context in a chain of reasoning which might take into account several axioms or premises of the form (1.2). Below, we shall formalize the accumulation of several contexts $c, d, \ldots$ into a composite context x by a generic operator $\oplus$, and specify the manner in which context accumulates by giving algebraic equations on $\oplus$ terms.

Some application areas where explicating the structure of context in this way promises to be useful are the following:

- Integrating heterogeneous data sources, such as in the management of knowledge bases distributed in the web; in this case the properties of the $\oplus$ operator will determine how two data sources are combined. Early work in this direction [21] has indicated the need for an algebraic structure, without explicating the algebraic-logical machinery to be employed.
- Understanding anaphora in natural language, where one approach [14] accumulates contextual information in decorated syntax trees, which invite elaboration by context-algebraic means.
- Interpreting fiction, where early work has left much to be desired. Taking fictions as separate contexts, and letting the way one fiction appears from within another be reflected in the properties of $\oplus$, work in progress [9] promises to improve the situation considerably.

Now we proceed to discuss rules for moving in and out of context. Consider these two rules for entering and exiting context, which use a notation reminiscent of Labelled Deductive Systems [7], and which bear a vague resemblance to the rule of necessitation and the inverse rule of necessitation in modal logic [6]:

$$\text{Enter: } \frac{\vdash x : ist(c, \lambda)}{\vdash x \oplus c : \lambda} \qquad\qquad \text{Exit: } \frac{\vdash x \oplus c : \lambda}{\vdash x : ist(c, \lambda)} \tag{1.3}$$

By the Enter rule, one may pass from $ist(c, \lambda)$ asserted in x, to λ asserted in $x \oplus c$. In the premise of this transition, the context of reasoning is x, and the asserted formula expresses that λ is true in context c. The result of the transition is that the new context of reasoning is $x \oplus c$, and the asserted formula is λ. In a sense, contextual information passes from the asserted formula into the surrounding context of reasoning. The new context of reasoning $x \oplus c$ expresses the combined context after, in context x, having additionally entered context c. In what follows, $x \oplus c$ will be modelled as an algebraic combination of x and c.

Vice versa, the premise of the Exit rule is that the context of reasoning is $x \oplus c$, and that λ is asserted there. The result is a new context of reasoning x, where it is asserted that λ is true in context c. Information is taken from the surrounding context of reasoning and put into the asserted formula. The context $x \oplus c$ has, so to speak, been split in two: a part c has been chipped off and used to express that λ is true there, and the residual part x is the new context of further reasoning. In our models, $x \oplus c$ will be an algebraic combination of x and c.

Taking this one step further, we consider an arbitrary series of interleaved Enter and Exit transitions. Upon entering a context it is desirable to leave a trace which shows the accumulated context after entering. A trace can have information about the context that is entered into, and/or of the surrounding context at the time.

Successive Enters augment the current context of reasoning with additional contextual information $b, c, \ldots$ by removing contextual items from formulas of the form $ist(b, ist(c, \ldots))$.

Conversely, on exiting from a context, the trace is picked up and modified for further reasoning purposes. Successive Exits construct nested formulas of the form $ist(d, ist(e, \ldots))$ by extracting parts $\ldots, e, d$ from the accumulated context of reasoning.

On this view then, the current context of reasoning can be compared to a data structure, which accumulates items of context during Enter transitions, and which releases items of context during Exit transitions.

We have purposely made no assumptions about the order and manner in which items of context are stored in the accumulated context of reasoning, and in coming sections we are going to develop the theory in general for a certain class of equationally specified disciplines, namely ground semigroups of context combination.

Let us compare this view with some contextual reasoning systems in the literature. In [4], we find a quantified logic of context where entering a context leaves a fairly uninformative trace, including only the last context that was entered:

$$\text{Enter: } \frac{\vdash x : ist(c, \lambda)}{\vdash c : \lambda} \qquad\qquad \text{Exit: } \frac{\vdash c : \lambda}{\vdash x : ist(c, \lambda)} \tag{1.4}$$

As another example, the accumulated context of reasoning could be the sequence of contexts entered into and not exited from, suggesting a stack discipline of reasoning. This is the approach taken in the propositional logic of context in [5], and corresponds to these rules:

$$\text{Enter: } \frac{\vdash \vec{c} : ist(c, \lambda)}{\vdash \vec{c}c : \lambda} \qquad\qquad \text{Exit: } \frac{\vdash \vec{c}c : \lambda}{\vdash \vec{c} : ist(c, \lambda)} \tag{1.5}$$

In the system we are about to develop, it is by no means prescribed that the contextual items taken from ist formulas in an Enter transition, will occur in any particular order during later Exits, or indeed that the same contextual items will occur at all. A particular algebra might change the contribution of one particular item of context if it occurs among certain others in a particular $\oplus$ term, for instance.

Along with these deduction rules the algebraic properties of the context constructor $\oplus$ are specified by equations. For example, the case of (1.5) corresponds to a purely associative constructor:

$$(u \oplus v) \oplus w = u \oplus (v \oplus w), \tag{1.6}$$

whereas (1.4) corresponds to a stronger condition on $\oplus$:

$$u \oplus v = v. \tag{1.7}$$

2 Overview

We now proceed to develop the theory for associative ground algebras of contexts. The paper is organized as follows: In the next section, we define algebras of contexts and the class of associative finite ground algebras. Then, the language of contextual formulas is defined, and some notational conventions established. Some options for semantical interpretation are discussed, arriving at a framework which defines truth of asserted formulas. Further, axioms schemata and deduction rules are put forward and discussed, and their soundness and completeness with respect to the semantical framework proved. A rule for exchanging algebraically equal contexts is defined and demonstrated to be sound, and a connection with substructural logic is pointed out. Then, a simplified presentation of the axiomatics is given, by a transformation of the algebraic equations. In this way, we obtain as special cases systems where the algebras are sets and bags (multisets), as well as the free semigroup (corresponding to (1.5)) and the flat semigroup (corresponding to (1.4)). The latter two correspond to systems described in [5] and [4]. In the concluding section, we point out some directions of further research.

3 Algebras of context

The algebraic machinery we need in this paper is very modest, having to do with a sort C for contexts, names for the elements of C, a binary operator $\oplus$, and ground equations.

In general algebraic terminology, a *signature* is a pair $\Sigma = \langle S, F\rangle$ where S is a set of *sorts* and F a set of *operators*, each with a *mapping type*: $F \to S^* \times S$. A set which provides elements for a sort of a signature is called a *carrier* for that sort. A Σ-*algebra* is an assignment of carriers and functions of correct type to the elements of S and F. Terms built from elements of F, and obeying the mapping types, are called Σ-*terms*, or F-*terms*, or *terms on* F. A pair of Σ-terms is called a Σ-*equation.* Such equations impose an equivalence relation on the set of ground Σ-terms, and the set of such equivalence classes gives rise to a special Σ-algebra. Given a set and an equivalence relation on that set, the set of equivalence classes is sometimes called the *quotient* set. For a full treatment of Σ-algebras with equations and formulas, consult e.g. [22].

Returning to our present purposes, we shall use the symbol C to denote both the sort of contexts and the set which is the carrier for that sort. So, let a countable set C of contexts be given a priori, and take it as the carrier of an algebra with an operator $\oplus$ of mapping type $\langle\langle C, C\rangle, C\rangle$, and equations

$$y_i = z_i, \quad 1 \le i \le N \tag{3.1}$$

for some $N > 0$, where y_i and z_i are terms on $\oplus$.

As an example, consider bags (multisets) over C, generated by the $\oplus$ operation. In a bag, as opposed to a sequence, the order of elements is immaterial. The relevant equations express associativity and commutativity of $\oplus$:

$$(u \oplus v) \oplus w \quad = \quad u \oplus (v \oplus w) \tag{3.2}$$

$$u \oplus v \quad = \quad v \oplus u \tag{3.3}$$

In a proper set over C, where repeated context entries don't count, the $\oplus$ operation is also idempotent:

$$u \oplus u \quad = \quad u. \tag{3.4}$$

We denote the set of all terms on $\oplus$ by $C^{\oplus}$, and the set of equivalence classes imposed by the algebraic equations we denote by $C^{\oplus}_{=}$.

It is sometimes convenient to include a special context ϵ, such that:

$$\epsilon \oplus u = u = u \oplus \epsilon. \tag{3.5}$$

Any desired properties of $\oplus$ are to be specified as algebraic equations in the axiomatics, including any or all of those mentioned here.

4 Associative Finite Ground Algebras

We proceed to treat a class of algebraic equations that provides a nice algebraic-logical correspondence. As an indication of what is in store here, compare the flatness schema of Buvač's quantificational logic of context [4]

$$\text{(Flat) } k : ist(k1, ist(k2, \phi)) \leftrightarrow ist(k2, \phi) \tag{4.1}$$

with the algebraic equation

$$x : u \oplus v = v. \tag{4.2}$$

It is tempting to ask whether all algebraic equations have an alternative presentation as axiom schemata expressing equivalence of certain *ist*-formulas. In this section, we establish criteria on the algebra which allow us to translate the algebraic equations into such schemata.

The contexts

$$k : k1 \ldots k2 \ldots k2$$

of schema (4.1) fit the pattern

$$x : u \ldots v \ldots v$$

of the algebraic equation (4.2), but there are a couple of caveats:

- Nesting of the *ist* predicate entails association of contexts into a sequence, so although $k1 \oplus k2$ suggests $ist(k1, ist(k2, \ldots))$, it is not straightforward to translate $k1 \oplus (k2 \oplus k3)$ into a nested *ist*-formula unless the algebra is associative.
- The *ist* predicate has only atomic contexts $k1, k2 \in C$ in its first coordinate, whereas u and v in the algebraic equation stand in for arbitrary context terms from $C^{\oplus}$.

We can steer clear of these obstacles by restricting the algebra, and this leads to a particularly neat axiomatic presentation of the corresponding systems.

Definition 4.1 (AFG algebras) An associative finite ground algebra, abbreviated AFG algebra, is one that satisfies these criteria:

- It is associative, i.e. contains the equation

$$\vdash x : (u \oplus v) \oplus w = u \oplus (v \oplus w).$$

- Every equation apart from associativity is restricted so that the variables in it can only be instantiated by constants, not by terms containing $\oplus$.

- The number of equations is finite.

A logic of context where the algebra is an AFG algebra, is called an AFG logic.

In AFG algebras, every equation other than associativity can be written

$$\vdash x : c_1 \oplus \ldots \oplus c_m = d_1 \oplus \ldots \oplus d_n \tag{4.3}$$

with $m, n > 0$, since parentheses are redundant because of associativity. The variables range over individual contexts.

From now on, we always assume that our algebra is an AFG algebra.

5 Formula language

Based on the discussion so far, we will use these language components:

- Disjoint sorts C for contexts, and T for other objects of discourse.
- Names (constants) for each of the elements of C and T.
- The $\oplus$ function symbol.
- Predicates $p(t_1, \ldots, t_m)$ on sorted coordinates, including:
- The identity predicate $t_1 = t_2$ for each sort.
- Truth-functional connectives $\neg, \rightarrow, \vee, \wedge, \leftrightarrow$.
- The *ist* modality.

For simplicity, our language will have no other (non-constant) function symbols than $\oplus$. Let therefore $\mathcal{L}$ be a sorted propositional language with identity, augmented by the special modality $ist(c, \lambda)$. The sorts will be C for contexts and T for other objects of discourse. More precisely, the language $\mathcal{L}$ of well-formed formulas can be defined in this way:

Definition 5.1 ($\mathcal{L}$, the set of well-formed formulas)

$$\mathcal{L} ::= \; P \mid \neg\mathcal{L} \mid \mathcal{L} \rightarrow \mathcal{L} \mid ist(C, \mathcal{L})$$

where P is a set of atomic predicates on sorted terms, including the identity predicate for each sort, and C is a set of context names.

Let us establish the following notational conventions:

- $\vee$, $\wedge$, and $\leftrightarrow$ are the usual abbreviations.
- Lower-case b, c, d, e denote constants or variables of sort C.
- Lower-case u, v, w, x, y, z denote context terms, i.e. elements of $C^{\oplus}$.
- Lower-case greek letters $\lambda, \chi, \mu, \ldots$ denote formulas from $\mathcal{L}$.
- Upper-case greek letters $\Omega, \Gamma, \ldots$ denote sets of formulas.
- $\perp$ denotes an arbitrary propositional contradiction, e.g. $\lambda \wedge \neg\lambda$.
- The result of replacing all substitutable occurrences of v in λ with u is denoted λ^v_u.
- $ist(c_1, c_2, \ldots, c_m, \lambda)$ is shorthand for $ist(c_1, ist(c_2, \ldots, ist(c_m, \lambda) \ldots))$.
  When $m \leq 0$, this is just λ.

- $\pm\phi$ denotes either ϕ or $\neg\phi$. If ϕ is an atomic formula, these are called *literals*.
- $ist^{\pm}(c_1, c_2, \ldots, c_m, \lambda)$ is shorthand for $\pm ist(c_1, \pm ist(c_2, \ldots, \pm ist(c_m, \lambda) \ldots))$, where the signs need not be all equal.
- $PROP$ is the set of ist-free formulae.

6 Semantical structure and truth conditions

We are on familiar ground as far as the propositional connectives are concerned. The critical aspect is the case where λ is $ist(c, \chi)$ for some c and χ, and we refer back to our discussion of Enter/Exit transitions: a model should assign the same truth value to $x : ist(c, \chi)$ as to $x \oplus c : \chi$. Now χ might in turn be $ist(d, \mu)$, leading to $x \oplus c \oplus d : \mu$, and so on.

We observed above how context is exchanged across the dividing colon during Enter and Exit transitions, and we now proceed to identify a subset of $C^{\oplus}$ which is going to be of frequent use later:

Definition 6.1 (x-continuants) For $x \in C^{\oplus}$, an x-continuant is any term $x \oplus c_1 \ldots \oplus c_m$, where $m \geq 0$ and $c_i \in C$ for $1 \leq i \leq m$.

Note that x itself is an x-continuant, because when $m = 0$ the condition $c_i \in C$ for $1 \leq i \leq m$ is vacuously true. It may well be that different x-continuants are equal by force of the algebraic equations:

$$x \oplus c_1 \ldots \oplus c_m = x \oplus d_1 \ldots \oplus d_n,$$

but this does not in general imply that

$$c_1 \oplus \ldots \oplus c_m = d_1 \oplus \ldots \oplus d_n$$

unless additional information about $\oplus$ is available.

The model structure below focuses on equivalence classes of x-continuants, so it's worth coining a term:

Definition 6.2 (x-bundles) For $x \in C^{\oplus}$, the set of x-bundles is the quotient of the set of x-continuants under the equivalence relation imposed by the algebraic equations.

The word 'bundle' was suggested by the thought of different linear strands bound together at both ends; all starting at x and all finally being equal.

We see that a model for asserted formulas of the form $x : \lambda$ must also be prepared to deal with $y : \mu$ whenever y is an x-continuant. But there is a complication: For x-continuants y, z that are equal by force of the algebraic equations, a model needs to assign the same truth value to

$$y : \mu \qquad \text{and} \qquad z : \mu$$

A model is therefore going to take x-bundles rather than x-continuants as arguments.

We take the view that a context can be vague, in the sense of encompassing a set of different possible circumstances. Any particular circumstance will be modelled as an interpretation of the formula language.

Thus, a model will associate each x-bundle to a set of interpretations. Intuitively, truth in a model of a formula asserted in a context requires truth at each interpretation bundled to that context.

Moreover, we do not commit to consistency of contexts, thus admitting the possibility of modelling contradictory contexts. Such a commitment can be made, if desired, by restricting the model structure so that the image of an x-bundle is always nonempty, and enriching the axiomatics correspondingly.

A point frequently made is that distinct contexts may well have different formula languages. This feature could be built into our system by mapping contexts to subsets of $\mathcal{L}$, and letting the semantical interpretation be three-valued. This technique was used in [5]. We believe this issue can be dealt with most naturally in a multi-language system in the style of [18], and it remains our ambition to endow multi-language systems with an algebraic superstructure reminiscent of the one we are discussing here. That will be in a future paper, however.

We are now ready to define the class of models for interpretation of asserted formulas. Let the set C of contexts and the set T of objects of discourse be given a priori. These are nonempty and no more than countable, and shall stay fixed throughout. Let the elements of T be rigid designators, i.e. let the constants of sort object of discourse correspond 1-1 to T, so that we may identify the set of constants of sort T with T itself.

Definition 6.3 (Rigid interpretations) A rigid interpretation is a first-order interpretation of the language, in which:

- the domain for objects of discourse is T,
- the domain for contexts is $C^{\oplus}_{=}$, the quotient of the set of $\oplus$ terms under the equivalence relation imposed by the algebraic equations,
- each constant of sort T is interpreted according to the 1-1 correspondence mentioned above,
- each constant of sort C is interpreted as its own equivalence class modulo the algebraic equations,
- the $\oplus$ symbol is interpreted homomorphically, i.e. $x \oplus y$ is interpreted as the set of terms $\widehat{x} \oplus \widehat{y}$ such that $\widehat{x}$ is in the set interpreting x and $\widehat{y}$ is in the set interpreting y,
- and the identity predicate for each sort is interpreted as the corresponding identity relation.

Definition 6.4 (x-models) An x-model is a function from x-bundles to sets of rigid interpretations.

Given an x-model M and an x-continuant y we shall frequently (in fact, usually) stretch the syntax a little and write $M(y)$ to denote the set of rigid interpretations that M associates with the x-bundle containing y.

We can now define truth and falsity of formulas asserted in context. Let $x \in C^{\oplus}, \lambda \in \mathcal{L}$, and let M be an x-model:

$$M \models x : \lambda \text{ iff } M, I \models x : \lambda \text{ for all } I \in M(x)$$

where for $I \in M(x)$

$$\begin{array}{lcll} M, I \models & x : p & \text{iff } I \text{ interprets } p \text{ as true} & (6.1) \\ M, I \models & x : \neg\lambda & \text{iff } M, I \not\models x : \lambda & \\ M, I \models & x : \lambda \to \gamma & \text{iff } M, I \models x : \lambda \ \ \text{implies } M, I \models x : \gamma & \\ M, I \models & x : ist(c, \lambda) & \text{iff } M, J \models x \oplus c : \lambda \ \ \text{for all } J \in M(x \oplus c) & \end{array}$$

Validity of an asserted formula is defined as truth in all models of the relevant context:

Definition 6.5 (Validity) We say that a formula λ is valid in context x, or synonymously that $x : \lambda$ is valid, in symbols

$$\models x : \lambda$$

iff $M \models x : \lambda$ for all x-models M.

7 Axiomatic presentation

The table gives axiom schemata and rules of deduction which are sound and complete with respect to the semantical framework.

The left and right sides, y_i resp. z_i, of each algebraic equation $x : y_i = z_i$, are terms from $C^{\oplus}$.

Reflexivity and congruence together imply symmetry and transitivity of the $=$ predicate, so we get all the familiar properties of equality.

PL and MP govern the the classical connectives.

K and Exit are the foundation of a normal multi-modal system.

The $\text{G}^{2,0,1,1}$ axiom and the axiom of nesting both have to do with x-continuant contexts. $\text{G}^{2,0,1,1}$ forces truth 'twice removed', so to speak, and later we shall see that $\text{G}^{2,0,1,1}$ generalizes to more deeply nested *ist* formulas, cfr (7.2). This axiom is a specialization and adaptation to the *ist* syntax of the modal schema

$$G^{k,l,m,n} \qquad \Diamond^k \Box^l A \to \Box^m \Diamond^n A$$

which is studied in [6].

The axiom of nesting relates bundles of x-continuants to equivalence of the correspondingly nested *ist* formulas. Note that x occurs on both sides of the colon in this schema, providing a connection between the formula being asserted and the context in which the assertion is made. In a later section, we'll see how the axiom of nesting can be subsumed by certain other axioms.

Definition 7.1 (Theoremhood) We say that a formula λ is derivable in context x, or synonymously that $x : \lambda$ is a theorem, in symbols

$$\vdash x : \lambda$$

iff $x : \lambda$ is an instance of an axiom schema, or follows from other theorems by applications of the deduction rules.

TABLE 1. **Axiom schemata and deduction rules**

Rules for changing context:

$$\text{Enter: } \frac{\vdash x : ist(c,\lambda)}{\vdash x \oplus c : \lambda} \qquad \text{Exit: } \frac{\vdash x \oplus c : \lambda}{\vdash x : ist(c,\lambda)}$$

Equational properties:

Reflexivity: $\vdash x : y = y$

Congruence: $\vdash x : y = z \rightarrow (\lambda^v_y \rightarrow \lambda^v_z)$

Algebraic equations: $\vdash x : y_i = z_i \qquad y_i, z_i \in C^{\oplus} \qquad 1 \le i \le N$

Propositional properties:

PL: $\vdash x : \lambda$ whenever λ is an instance of a propositional tautology

$$\text{MP: } \frac{\vdash x : \lambda \quad \vdash x : \lambda \rightarrow \chi}{\vdash x : \chi}$$

Modal properties:

K: $\vdash x : ist(c, \lambda \rightarrow \chi) \rightarrow (ist(c,\lambda) \rightarrow ist(c,\chi))$

$\mathrm{G}^{2,0,1,1}$: $\vdash x : \neg ist(c, ist(d,\lambda)) \rightarrow ist(c, \neg ist(d,\lambda))$

Nesting: $\vdash x : x \oplus c_1 \ldots \oplus c_m = x \oplus d_1 \ldots \oplus d_n \rightarrow (ist(c_1,\ldots,c_m,\lambda) \rightarrow ist(d_1,\ldots,d_n,\lambda))$

Derived schemata:

Let us note that the following useful schemata are derivable. For $n > 0$:

$$\mathrm{K}^n : \quad \vdash x : ist(c_1,\ldots,c_n,\lambda \rightarrow \chi) \rightarrow (ist(c_1,\ldots,c_n,\lambda) \rightarrow ist(c_1,\ldots,c_n,\chi)) \tag{7.1}$$

$$\mathrm{G}^{n+1,0,n,1} : \quad \vdash x : \neg ist(c_1,\ldots,c_{n+1},\lambda) \rightarrow ist(c_1,\ldots,c_n,\neg ist(c_{n+1},\lambda)) \tag{7.2}$$

$$\mathrm{AND}^n : \quad \vdash x : ist(c_1,\ldots,c_n,\lambda \wedge \chi) \leftrightarrow ist(c_1,\ldots,c_n,\lambda) \wedge ist(c_1,\ldots,c_n,\chi) \tag{7.3}$$

$$\mathrm{OR}^n : \quad \vdash x : ist(c_1,\ldots,c_n,\lambda) \vee ist(c_1,\ldots,c_n,\chi) \rightarrow ist(c_1,\ldots,c_n,\lambda \vee \chi) \tag{7.4}$$

$$\Delta^{\pm} : \quad \vdash x : ist(c, \pm ist(d,\lambda) \vee \chi) \rightarrow ist(c, \pm ist(d,\lambda)) \vee ist(c,\chi) \tag{7.5}$$

The proofs are included in an appendix. Note that $\mathrm{G}^{1,0,0,1}$ is not a theorem of this system:

$$\not\vdash x : \neg ist(c, \lambda) \rightarrow ist(c, \neg\lambda).$$

The case $n = 1$ of schemas (7.3) and (7.4) will be referred to as AND and OR, respectively, without superscripts. In schema (7.5), the sign of $\pm ist(d, \lambda)$ is the same on both sides of the implication.

8 Soundness

We verify that the axiom schemata are semantically valid and that the deduction rules preserve semantic validity, formally in the case of $G^{2,0,1,1}$ and by brief comments and remarks for the rest.

The Enter/Exit rules are easily seen to be valid from the *ist* clause of the model conditions. Reflexivity and congruence are valid by the rigidity requirements. As for nesting, the premise places the two x-continuants in the same x-bundle, and validity follows.

In the algebraic equations, the terms y_i resp. z_i are interpreted as the same equivalence class of terms by the rigidity requirements, and validity follows by rigidity of identity interpretation.

PL is valid because the model conditions respect the truth-functional connectives, and MP preserves validity for the same reason. Also, K is valid since for any x-model M and any context c, M also respects the truth-functional connectives when considered as an $x \oplus c$-model.

Concerning $G^{2,0,1,1}$, take any x-model M and an interpretation $I \in M(x)$ and assume that $M, I \models x : \neg ist(c, ist(d, \lambda))$. It then follows that $M, J \not\models x \oplus c : ist(d, \lambda)$ for some $J \in M(x \oplus c)$, and that $M, H \not\models x \oplus c \oplus d : \lambda$ for some $H \in M(x \oplus c \oplus d)$.

For validity of $G^{2,0,1,1}$, we now require $M, I \models x : ist(c, \neg ist(d, \lambda))$, which is equivalent to $M, Q \models x \oplus c : \neg ist(d, \lambda)$ for every $Q \in M(x \oplus c)$, which is equivalent to $M, R \not\models x \oplus c \oplus d : \lambda$ for some $R \in M(x \oplus c \oplus d)$ for every such $Q \in M(x \oplus c)$. By the argument in the preceding paragraph, H fills the requirement for such an R.

9 Completeness

We begin by setting out definitions of consistency and maximality indexed by contexts.

Definition 9.1 (x-consistency) Let $x \in C^{\oplus}$.

- a formula λ is x-consistent iff $\not\vdash x : \neg\lambda$.
- a finite set of formulas is x-consistent iff their conjunction is x-consistent
- an infinite set of formulas is x-consistent iff every finite subset is x-consistent.

Definition 9.2 (x-maximality) Let $x \in C^{\oplus}$.

A set Λ of formulas is x-maximal iff it is x-consistent and for every formula λ, $\Lambda \cup \{\lambda\}$ is x-consistent only if $\lambda \in \Lambda$.

The properties of x-maximal sets are the familiar ones, cfr. e.g. [6, 13]. We shall construct x-maximal sets in a way that reflects our axiomatics:

Lemma 9.3 (cfr. Lindenbaum's lemma) Every x-consistent set of formulas can be extended to a x-maximal set.

Proof. Take a set Γ_0 of x-consistent formulas and an enumeration of all formulas $\mathcal{L} = \{\gamma_1, \gamma_2, \ldots\}$, and construct Γ_i from Γ_{i-1}, $i > 0$, as follows: If $\Gamma_{i-1} \cup \{\gamma_i\}$ is x-consistent, let

$$\Gamma_i = \Gamma_{i-1} \cup \{\gamma_i\},$$

otherwise let

$$\Gamma_i = \Gamma_{i-1}.$$

Every Γ_i is x-consistent, because every addition of a formula is made x-consistently. The union

$$\Gamma = \bigcup_{i=0}^{\infty} \Gamma_i$$

is x-consistent, because for every finite subset Γ' of Γ with $\gamma_j \in \Gamma'$ being the element of highest index, we have $\Gamma' \subseteq \Gamma_j$, which is x-consistent. Γ is also x-maximal, since $\Gamma \cup \{\gamma_i\}$ is x-consistent iff $\gamma_i \in \Gamma_i \subset \Gamma$. ■

Now we fix some context x and an x-consistent formula δ, and embark on the construction of an x-model for the asserted formula $x : \delta$. First we need an x-maximal set of formulas containing δ:

Definition 9.4 (Ω, an x-maximal extension of $\{\delta\}$) Let Ω be a set of formulas which is x-maximal, extending the set $\{\delta\}$, and constructed as in the proof of lemma 9.3.

The set Ω, being x-maximal, can be construed as a complete account of truth in context x. For every theorem $\vdash x : \lambda$, $\lambda \in \Omega$, and for every $\lambda \in \mathcal{L}$, either $\lambda \in \Omega$ or $\neg\lambda \in \Omega$. This is but one out of several possible accounts, since the construction in lemma 9.3 is not uniquely specified.

Ω also gives partial accounts of truth in x-continuant contexts, by virtue of nested *ist*$(\ldots,\ldots)$ formulas contained in it. By retracting the nesting of certain *ist* formulas in Ω, we define some formula sets which are going to be useful in defining an x-model for δ:

Definition 9.5 (Ω retracts) For contexts $c_1, \ldots, c_n \in C$, let

$$\Omega_{c_1 \ldots c_n} = \{ \quad \phi \quad | \quad ist(c_1, \ldots, c_n, \phi) \in \Omega \text{ and } \phi \in PROP\}.$$

Retracts of Ω are not necessarily x-maximal, and they may even be x-inconsistent. Never the less, if an Ω retract is x-consistent, it gives a partial account of truth in the corresponding x-continuant context, and if so it gives rise to a set of rigid interpretations which agree with this partial account. Those interpretations will define the behaviour of our x-model of δ on the corresponding x-continuant.

We are now in a position to define an x-model M such that $M \models x : \delta$, which in turn will prove completeness.

Definition 9.6 (M, a model of $x : \delta$) Let the mapping M from x-bundles to sets of rigid interpretations be defined as follows:

Whenever $\widehat{x}$ is an x-bundle containing $x \oplus c_1 \ldots \oplus c_m$, let

$$M(\widehat{x}) = \{ \quad I \quad | \quad I \text{ is rigid and validates every formula in } \Omega_{c_1 \ldots c_m}\}.$$

M is well-defined because of the following lemma:

Lemma 9.7 (Congruence of retracts) If $\vdash x : x \oplus c_1 \ldots \oplus c_m = x \oplus d_1 \ldots \oplus d_n$, then $\Omega_{c_1 \ldots c_m} = \Omega_{d_1 \ldots d_n}$

Proof. Suppose the premise is true, and let $\lambda \in \Omega_{c_1 \ldots c_m}$. Then by construction of retracts $ist(c_1, \ldots, c_m, \lambda) \in \Omega$, and by x-maximality of Ω and the axiom of nesting it follows that $ist(d_1, \ldots, d_n, \lambda) \in \Omega$, and hence by definition of retracts $\lambda \in \Omega_{d_1 \ldots d_n}$. The converse is symmetric. ∎

The proof of completeness hinges on the fact that the x-model M models a formula asserted in a certain x-continuant context if and only if that formula is a member of the corresponding Ω retract. To help with the proof of this, let us see that there is a bonus to be had if an Ω retract is x-consistent. Recall that $\perp$ means any propositional contradiction, so no x-consistent set contains $\perp$.

Lemma 9.8 (D approximation) For $j > 0$ and $c_1, \ldots, c_j \in C$ and $\lambda \in \mathcal{L}$, if $\neg ist(c_1, \ldots, c_{j-1}, \perp) \in \Omega$, then

$$ist^{\pm}(c_1, \ldots, c_j, \lambda) \in \Omega \quad \text{iff} \quad \pm\, ist(c_1, \ldots c_j, \lambda) \in \Omega$$

where the sign on the right is positive iff the number of negations in the $ist^{\pm}$ on the left is even.

Proof. The proof is by induction on the number of negations in the $ist^{\pm}$ on the left hand side. In the base case the number is 0, and the result is trivial. For the inductive step, let i be the index of the first negation on the left hand side: $ist(c_1, \ldots, c_{i-1}, \neg ist^{\pm}(c_i, \ldots, c_j, \lambda)) \in \Omega$. If $i = 1$, the result is immediate by induction and x-maximality of Ω, otherwise we have $1 < i \leq j$. We start by proving

$$ist(c_1, \ldots, c_{i-1}, \neg ist(c_i, \psi)) \in \Omega \quad \text{iff} \quad \neg ist(c_1, \ldots, c_{i-1}, ist(c_i, \psi)) \in \Omega \quad (9.1)$$

where ψ is the subformula $ist^{\pm}(c_{i+1}, \ldots, c_j, \lambda)$ of $ist^{\pm}(c_1, \ldots, c_j, \lambda)$. From the premise $\neg ist(c_1, \ldots, c_{j-1}, \perp) \in \Omega$ we have in particular $\neg ist(c_1, \ldots, c_{i-1}, \perp) \in \Omega$, and we can replace $\perp$ by the contradiction

$$\neg ist(c_i, \psi) \wedge ist(c_i, \psi),$$

obtaining

$$\neg ist(c_1, \ldots, c_{i-1}, \neg ist(c_i, \psi) \wedge ist(c_i, \psi)) \in \Omega,$$

and by x-maximality of Ω

$$ist(c_1, \ldots, c_{i-1}, \neg ist(c_i, \psi) \wedge ist(c_i, \psi)) \notin \Omega.$$

Since $i > 1$, we obtain by AND^{i-1} and x-maximality of Ω

$$(ist(c_1, \ldots, c_{i-1}, \neg ist(c_i, \psi)) \wedge ist(c_1, \ldots, c_{i-1}, ist(c_i, \psi))) \notin \Omega,$$

which, again by x-maximality, amounts to

$$(ist(c_1,\ldots,c_{i-1},\neg ist(c_i,\psi)) \rightarrow \neg ist(c_1,\ldots,c_{i-1},ist(c_i,\psi))) \in \Omega,$$

or equivalently

$$\text{if } ist(c_1,\ldots,c_{i-1},\neg ist(c_i,\psi)) \in \Omega \text{ then } \neg ist(c_1,\ldots,c_{i-1},ist(c_i,\psi))) \in \Omega.$$

This shows the left-to-right direction of (9.1). The other direction follows by $\mathrm{G}^{i,0,i-1,1}$ and x-maximality of Ω, so we see that the first negation sign has been "bubbled out". This can be repeated for each of the remaining negations, proving the lemma. ∎

We can now prove that M is an x-model of the members of Ω, from which completeness will follow. The structure of the proof is adapted from [5], and uses the fact that every formula is provably equivalent to one in a specific normal form:

Definition 9.9 (CNF) A formula ϕ is in conjunctive normal form (CNF) iff it is of the form $\eta_1 \wedge \eta_2 \wedge \ldots \wedge \eta_p$, and each η_i is of the form $\alpha_{i1} \vee \alpha_{i2} \vee \ldots \vee \alpha_{ir_i}$, where each α_{ij} is either a literal or $ist^{\pm}(c_1,\ldots,c_m,\beta)$ for some disjunction of literals β. Here, p and r_i can be 1.

Lemma 9.10 (CNF) For any context $x \in C^{\oplus}$ and any formula $\phi \in \mathcal{L}$, there exists a formula ϕ^* which is in CNF, such that $\vdash x : \phi \leftrightarrow \phi^*$.

Proof. The proof is by induction on the syntactical structure of ϕ.

- The base case when ϕ is an atom is trivial, with ϕ itself as ϕ^*, since $\vdash x : \phi \leftrightarrow \phi$ is an instance of axiom schema PL.
- When ϕ is $\mu_1 \wedge \mu_2$, we have by induction μ_1^* and μ_2^* both in CNF, such that $\vdash x : \mu_1 \leftrightarrow \mu_1^*$ and $\vdash x : \mu_2 \leftrightarrow \mu_2^*$. Then $\mu_1^* \wedge \mu_2^*$ is in CNF, and $\vdash x : \phi \leftrightarrow \mu_1^* \wedge \mu_2^*$.
- When ϕ is $\mu_1 \vee \mu_2$, again we have by induction μ_1^* and μ_2^* both in CNF, such that $\vdash x : \mu_1 \leftrightarrow \mu_1^*$ and $\vdash x : \mu_2 \leftrightarrow \mu_2^*$. We have $\vdash x : \phi \leftrightarrow \mu_1^* \vee \mu_2^*$, but the right hand side is not necessarily in CNF. Now we note that the propositional laws of distribution of disjunction over conjunction are instances of axiom schema PL:

  $$\begin{array}{lll} \vdash & x: & (\alpha \wedge \beta) \vee \gamma \leftrightarrow (\alpha \vee \gamma) \wedge (\beta \vee \gamma) \\ \vdash & x: & \alpha \vee (\beta \wedge \gamma) \leftrightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \end{array}$$

  By repeated application of these, we obtain a formula μ_3^* in CNF, such that $\vdash x : \mu_1^* \vee \mu_2^* \leftrightarrow \mu_3^*$, yielding $\vdash x : \phi \leftrightarrow \mu_3^*$ as required.
- When ϕ is $\neg\mu$ for some μ, we have by induction μ^* in CNF such that $\vdash x : \mu \leftrightarrow \mu^*$, and $\vdash x : \phi \leftrightarrow \neg\mu^*$, but $\neg\mu^*$ is not necessarily in CNF. Now we note that also the distribution and cancellation laws for negation are instances of axiom schema PL:

  $$\begin{array}{lll} \vdash & x: & \neg(\alpha \wedge \beta) \leftrightarrow (\neg\alpha \vee \neg\beta) \\ \vdash & x: & \neg(\alpha \vee \beta) \leftrightarrow (\neg\alpha \wedge \neg\beta) \\ \vdash & x: & \neg\neg\alpha \leftrightarrow \alpha \end{array}$$

  Application of these, along with the previously listed distribution laws, yields a formula λ in CNF such that $\vdash x : \neg\mu^* \leftrightarrow \lambda$, and this leads to $\vdash x : \phi \leftrightarrow \lambda$, as required.

- When ϕ is $ist(c, \mu)$ we have by induction μ^* in CNF such that $\vdash x \oplus c : \mu \leftrightarrow \mu^*$. By application of K, Exit, and MP, this gives $\vdash x : \phi \leftrightarrow ist(c, \mu^*)$, but $ist(c, \mu^*)$ is not necessarily in CNF. We can write

  $$\mu^* = \bigwedge_{i=1}^{p} \bigvee_{j=1}^{r_i} \alpha_{ij}$$

  and we can assume without loss of generality the existence of indices q_i such that α_{ij} is of the form $ist^{\pm}(c_1, \ldots, c_m, \beta)$ for some disjunction of literals β when $1 \leq j \leq q_i$, and that α_{ij} is a literal when $q_i < j \leq r_i$. Either interval can be empty. By repeated application of the AND schema, along with PL and MP, we obtain

  $$\vdash x : \phi \leftrightarrow \bigwedge_{i=1}^{p} ist(c, \bigvee_{j=1}^{r_i} \alpha_{ij}).$$

  Now we apply OR and $\Delta^{\pm}$, along with PL and MP, q_i times for each i, to distribute the i'th ist over the first q_i disjunctions, obtaining

  $$\vdash x : \phi \leftrightarrow \bigwedge_{i=1}^{p} \left(\bigvee_{j=1}^{q_i} ist(c, \alpha_{ij}) \bigvee ist(c, \bigvee_{j=q_i+1}^{r_i} \alpha_{ij}) \right),$$

  and here the right hand side is in CNF as required.

■

Lemma 9.11 (Autovalidation) For $\chi \in \mathcal{L}$

$$\chi \in \Omega \text{ iff } M \models x : \chi$$

Proof. By Lemma 9.10 it is sufficient to consider formulas χ in CNF. The proof is by induction on the syntactical structure of χ.

- In the base case, χ is an atomic formula, and thus in CNF. The lemma follows because M has been constructed so as to map x to the set of interpretations that validate all propositional formulas in Ω.
- For the inductive case where χ is $\neg\mu$ for some μ, notice that if χ is in CNF, then so is μ. Now, $\chi \in \Omega$ iff, by x-maximality, $\mu \notin \Omega$, iff, by inductive hypothesis, $M \not\models x : \mu$, iff, by the model conditions, $M \models x : \chi$.
- When χ is $\mu_1 \vee \mu_2$ for some μ_1 and μ_2, again notice that if χ is in CNF, then so are μ_1 and μ_2. Now, $\chi \in \Omega$ iff, by x-maximality, $\mu_1 \in \Omega$ or $\mu_2 \in \Omega$, iff, by inductive hypothesis, $M \models x : \mu_1$ or $M \models x : \mu_2$, iff, by the model conditions, $M \models x : \mu_1 \vee \mu_2$.
- The inductive case for conjunction is similar.
- The case where χ is an ist-formula is the core of the matter. On the assumption that χ is in CNF, we can write

  $$\chi = ist^{\pm}(c_1, \ldots, c_m, \mu)$$

  with μ a disjunction of literals. The proof proceeds in two cases, depending on whether or not any of the retracts $\Omega_{c_1 \ldots c_j}$ with $j < m$ are inconsistent:

- In the first case, $\neg ist(c_1, \ldots, c_{m-1}, \bot) \in \Omega$. Now by D approximation, $ist^{\pm}(c_1, \ldots, c_m, \mu) \in \Omega$ iff $\pm ist(c_1, \ldots, c_m, \mu) \in \Omega$, where the latter sign is positive iff the parity of signs in $ist^{\pm}(c_1, \ldots, c_m, \mu)$ is even. The proof now proceeds by cases of even and odd parity:
  - ∗ Even parity: On the one hand, $ist^{\pm}(c_1, \ldots, c_m, \mu) \in \Omega$ iff, by D approximation, $ist(c_1, \ldots, c_m, \mu) \in \Omega$ iff, by definition of retracts, $\mu \in \Omega_{c_1 \ldots c_m}$. On the other hand, with even parity of negations, $M \models x : ist^{\pm}(c_1, \ldots, c_m, \mu)$ is equivalent to $M, I \models x \oplus c_1 \ldots \oplus c_m : \mu$ for all $I \in M(x \oplus c_1 \ldots \oplus c_m)$ by the model conditions. So, we prove that

    $$\mu \in \Omega_{c_1 \ldots c_m} \text{ iff } M, I \models x \oplus c_1 \ldots \oplus c_m : \mu \text{ for all } I \in M(x \oplus c_1 \ldots \oplus c_m).$$

    The left-to-right direction is immediate from the construction of M. For the right-to-left direction, suppose for contradiction that $\mu \notin \Omega_{c_1 \ldots c_m}$, which by construction of retracts means that $ist(c_1, \ldots, c_m, \mu) \notin \Omega$, and by x-maximality of Ω that $\neg ist(c_1, \ldots, c_m, \mu) \in \Omega$. Then by $\mathrm{G}^{m+1,0,m,1}$ we get $ist(c_1, \ldots, c_m, \neg\mu) \in \Omega$, from which we have $\neg\mu \in \Omega_{c_1 \ldots c_m}$ since $\neg\mu \in PROP$. Therefore no interpretation I which validates all of $\Omega_{c_1 \ldots c_m}$ can validate μ, and this contradicts $M, I \models x \oplus c_1 \ldots \oplus c_m : \mu$.
  - ∗ Odd parity: This case reduces to the even parity case because

    $$\neg ist(c_1, \ldots, c_m, \mu) \in \Omega \text{ iff } M \models x : \neg ist(c_1, \ldots, c_m, \mu)$$

    is equivalent to

    $$ist(c_1, \ldots, c_m, \mu) \in \Omega \text{ iff } M \models x : ist(c_1, \ldots, c_m, \mu)$$

    by x-maximality of Ω and the model conditions.
- Otherwise let j be the index of the first inconsistent Ω retract with $j < m$. Now we have $\neg ist(c_1, \ldots, c_{j-1}, \bot) \in \Omega$, and $ist(c_1, \ldots, c_j, \bot) \in \Omega$, and $M(x \oplus c_1 \ldots \oplus c_j) = \emptyset$. Furthermore, by D approximation:

  $$ist^{\pm}(c_1, \ldots, c_m, \mu) \in \Omega \text{ iff } \pm ist(c_1, \ldots, c_j, ist^{\pm}(c_{j+1}, \ldots, c_m, \mu)) \in \Omega,$$

  where the sign on the right hand side is positive iff the parity of negations in the first j indices of χ is even. The proof proceeds by even and odd parity of negations in the first j indices:
  - ∗ Even parity: We show that $\chi \in \Omega$ and $M \models x : \chi$ are both true.

    $$ist(c_1, \ldots, c_j, ist^{\pm}(c_{j+1}, \ldots, c_m, \mu)) \in \Omega$$

    is true because $ist(c_1, \ldots, c_j, \bot) \in \Omega$ and Ω is x-maximal, and

    $$M \models x : ist(c_1, \ldots, c_j, ist^{\pm}(c_{j+1}, \ldots, c_m, \mu))$$

    is true because $M(x \oplus c_1 \ldots \oplus c_j) = \emptyset$. With even parity of negations among the first j indices in χ, this is equivalent to $M \models x : \chi$ by the model conditions.
  - ∗ Odd parity: This case reduces to the even parity case in the same way as before.

■

Theorem 9.12 (Completeness) $\models x : \phi$ only if $\vdash x : \phi$

Proof. Every x-consistent formula has an x-model, by lemma 9.11 with $\chi = \phi$. ■

10 Exchanging equal contexts of reasoning

Let us see that it is safe to replace the surrounding context of reasoning with one that is equal according to the algebra, by proving that the following inference rule is semantically sound:

$$\text{RC:} \quad \frac{\vdash x : y = z \qquad \vdash y : \lambda}{\vdash z : \lambda} \tag{10.1}$$

It expresses that wherever reasoning is taking place and the current context is algebraically equal to another context, then the current context can be exchanged for the other, equal context. In general, y and z can be syntactically different terms on $\oplus$, but belonging to the same equivalence class.

In view of completeness, (10.1) can be rephrased as:

$$\text{If } \models x : y = z \text{ and } \models y : \lambda, \text{ then } \models z : \lambda. \tag{10.2}$$

To verify this, we need a lemma to the effect that algebraically equal contexts have the same bundles of continuants:

Lemma 10.1 (Congruence of bundles) If $\vdash x : y = z$
then $\vdash x : y \oplus c_1 \ldots \oplus c_m = z \oplus c_1 \ldots \oplus c_m$.

Proof. The proof is direct from Reflexivity, Congruence, and Modus Ponens:

1 $\vdash x : y = z$ Hypothesis

2 $\vdash x : y = z \to$
$(y \oplus c_1 \ldots \oplus c_{m-1} = y \oplus c_1 \ldots \oplus c_{m-1} \to y \oplus c_1 \ldots \oplus c_{m-1} = z \oplus c_1 \ldots \oplus c_{m-1})$
Congruence

3 $\vdash x : (y \oplus c_1 \ldots \oplus c_{m-1} = y \oplus c_1 \ldots \oplus c_{m-1} \to y \oplus c_1 \ldots \oplus c_{m-1} = z \oplus c_1 \ldots \oplus c_{m-1}$
1, 2, MP

4 $\vdash x : y \oplus c_1 \ldots \oplus c_m = z \oplus c_1 \ldots \oplus c_m$ Reflexivity, 3, MP

■

Now we are ready to exchange algebraically equal contexts of reasoning. Instead of (10.2) we shall prove the following stronger statement:

Theorem 10.2 If $\models x : y = z$, then for every y-model M (which is then also a z-model), and every $I \in M(y)$ (hence also $I \in M(z)$), and every formula λ, $M, I \models y : \lambda$ iff $M, I \models z : \lambda$

Proof. by induction on the structure of λ

$\lambda \in P$: I validates λ iff I validates λ, trivially.

λ is $\neg\mu$: $M, I \not\models y : \mu$ iff $M, I \not\models z : \mu$ by the inductive hypothesis.

λ is $\mu_1 \to \mu_2$: Suppose $M, I \models y : \mu_1 \to \mu_2$ and $M, I \models z : \mu_1$. By the inductive hypothesis the latter is equivalent to $M, I \models y : \mu_1$, and by the model conditions it follows that $M, I \models y : \mu_2$, which by the inductive hypothesis is equivalent to $M, I \models z : \mu_2$. The argument in the other direction is symmetric.

λ is $ist(c, \mu)$: We have $J \in M(y \oplus c)$ iff $J \in M(z \oplus c)$ by lemma 10.1, so we get $M, J \models y \oplus c : \mu$ iff $M, J \models z \oplus c : \mu$ by the inductive hypothesis.

■

11 Subsuming the axiom of nesting by AFG axioms

We are going to let each equation of the form (4.3):

$$\vdash x : c_1 \oplus \ldots \oplus c_m = d_1 \oplus \ldots \oplus d_n$$

give rise to a corresponding axiom schema:

$$\vdash x : ist(c_1, \ldots, c_m, \lambda) \leftrightarrow ist(d_1, \ldots, d_n, \lambda) \tag{11.1}$$

which follows by associativity, congruence, nesting and MP. The result is that the axiom of nesting is subsumed by the axioms of the form (11.1), in the following sense:

Theorem 11.1 (Subsumption of Nesting) In an AFG logic, where the equations of the form (4.3) have been replaced by corresponding axioms of the form (11.1), for any ground instance of the axiom of nesting, whenever the equality in the premise of the axiom of nesting is true, the conclusion of same instance of the axiom of nesting is provable from (11.1).

Proof. To see this, we first notice that the following deduction rule is admissible:

$$\text{RRI:} \quad \frac{\vdash x : ist(c, \lambda) \rightarrow ist(c, \chi)}{\vdash x \oplus c : \lambda \rightarrow \chi} \tag{11.2}$$

In an AFG logic, any instance of the premise of the axiom of nesting:

$$x \oplus b_1 \ldots \oplus b_m = x \oplus e_1 \ldots \oplus e_m \tag{11.3}$$

can be written

$$a_1 \oplus \ldots \oplus a_k \oplus b_1 \oplus \ldots \oplus b_m = a_1 \oplus \ldots \oplus a_k \oplus e_1 \oplus \ldots \oplus e_n \tag{11.4}$$

where x is $a_1 \oplus \ldots \oplus a_k$ for context names $a_j, 1 \leq j \leq k$.

Because of rigidity of the $=$ predicate, if such a premise is true anywhere, it is true everywhere, and by completeness it is then also a theorem asserted in any context. So, given a theorem corresponding to the premise of the axiom of nesting:

$$\vdash x : a_1 \oplus \ldots \oplus a_k \oplus b_1 \oplus \ldots \oplus b_m = a_1 \oplus \ldots \oplus a_k \oplus e_1 \oplus \ldots \oplus e_n \tag{11.5}$$

we can prove

$$\vdash x : ist(b_1, \ldots, b_m, \lambda) \rightarrow ist(e_1, \ldots, e_n, \lambda) \tag{11.6}$$

from the axioms of the form (11.1), without using the axiom of nesting. There will then be a chain of equal terms $y_0 = \ldots = y_h$ such that

$$y_0 = a_1 \oplus \ldots \oplus a_k \oplus b_1 \oplus \ldots \oplus b_m$$

$$y_h = a_1 \oplus \ldots \oplus a_k \oplus e_1 \oplus \ldots \oplus e_n$$

and

$$\vdash x : y_{i-1} = y_i, 1 \leq i \leq h$$

because of an AFG equation of the form (4.3) and congruence. A proof of (11.6) is then obtained by a corresponding chain of inferences from this tautology

$$\vdash \epsilon : ist(a_1, \ldots, a_k, b_1, \ldots, b_m, \lambda) \rightarrow ist(a_1, \ldots, a_k, b_1, \ldots, b_m, \lambda)$$

using the corresponding AFG axioms of the form (11.1) and congruence, resulting in

$$\vdash \epsilon : ist(a_1, \ldots, a_k, b_1, \ldots, b_m, \lambda) \rightarrow ist(a_1, \ldots, a_k, e_1, \ldots, e_n, \lambda)$$

followed by k applications of RRI, ending in

$$\vdash x : ist(b_1, \ldots, b_m, \lambda) \rightarrow ist(e_1, \ldots, e_n, \lambda).$$

The derivation of the converse implication is symmetric.

■

On the strength of this, an AFG logic can be presented by taking associativity, as well as the algebraic equations and the induced equivalences on context terms, for granted, presenting the $\oplus$ operation, rigid interpretations, x-bundles and x-models in the terminology of the chosen algebra, giving axiom schemata of the form (11.1) in place of equations of the form (4.3) and deleting the axiom of nesting.

The axioms and deduction rules common to all AFG systems are therefore the ones of table 1, except the axiom of nesting, and with the addition of axioms of the form (11.1) corresponding to the AFG equations in each case. In the subsections that follow, we show how this yields some context systems from the literature, as well as some new ones, by choosing different AFG algebras. For each one, we display the Enter/Exit rules, and the special axioms of the form (11.1).

11.1 Context sequences

The simplest option is to leave the algebra as purely associative, with no additional equations of the form (4.3). This expresses the intuition that changes to the surrounding context follow a LIFO discipline. This option is implicit in the propositional logic of context of Buvač, Buvač, and Mason [5].

The $\oplus$ operation is then concatenation of finite sequences, denoted by juxtaposition. Each context name is identified with the singleton sequence containing itself. The equation expressing associativity is taken as implicit. An x-continuant is simply a sequence starting with x, and the concepts of x-bundles and x-continuants coincide. Identity of sequences is pairwise identity of elements.

The model structure is formulated as a mapping from the set of finite sequences of contexts, denoted C^*, to sets of rigid first-order interpretations, which corresponds exactly to taking for granted the equivalence classes imposed by associativity alone.

This is the form of the Enter/Exit rules in purely associative AFG algebras:

$$\text{Enter: } \frac{\vdash x : ist(c, \lambda)}{\vdash xc : \lambda} \qquad \text{Exit: } \frac{\vdash xc : \lambda}{\vdash x : ist(c, \lambda)} \qquad \text{where } x \in C^*, c \in C.$$

There are no explicit AFG axioms in this case, only associativity of $\oplus$ which is built-in.

In [5], a propositional logic of contexts is presented where the semantics is based on sequences. Its axioms and rules are equivalent to those given here.

11.2 Context multisets

When dealing with independent contexts, it is reasonable to think that entering one context and then another should amount to the same as entering them in the opposite order. This corresponds to thinking about the surrounding context as a multiset, i.e. a collection where the order of the constituents is immaterial. This is expressed algebraically by commutativity of $\oplus$:

$$\vdash x : u \oplus v = v \oplus u.$$

We observe that multiset equality can be expressed by associativity in conjunction with commutativity restricted to context constants:

$$\vdash x : c \oplus d = d \oplus c \tag{11.7}$$

because with associativity we can get any permutation of

$$c_1 \oplus \ldots \oplus c_m$$

by a series of applications of (11.7) on adjacent elements.

Thus, we have an AFG presentation of multisets, and we get a logic of multisets of contexts by taking models to be mappings from multisets of contexts to sets of rigid first-order interpretations, taking $\oplus$ to be multiset union $\sqcup$ and identifying each context name c with the multiset $[c]$ containing only a single occurrence of itself, and taking context collections to be identical if they are equal as multisets.

This is therefore the form of the Enter/Exit rules for multisets (bags):

$$\text{Enter: } \frac{\vdash x : ist(c, \lambda)}{\vdash x \sqcup [c] : \lambda} \qquad \text{Exit: } \frac{\vdash x \sqcup [c] : \lambda}{\vdash x : ist(c, \lambda)} \qquad \text{where } x \in \text{bag}(C), c \in C.$$

and the following AFG axiom schema expresses the necessary additional property of multiset continuants, namely commutativity:

$$\text{COMM:} \quad \vdash x : ist(c, d, \lambda) \leftrightarrow ist(d, c, \lambda).$$

The Hilbert system for context multisets is obtained from the common AFG rules and axioms by revising the Enter/Exit rules as shown here, and adding axiom COMM.

11.3 Context sets

If one feels that a particular item of context makes the same contribution to the accumulated outer context whether it is entered into once or more than once, then it is reasonable to say that context composition is idempotent:

$$\vdash x : u \oplus u = u. \tag{11.8}$$

If taken in conjunction with commutativity as in the preceding subsection, the resulting context algebra is that of sets. Idempotence can then be adequately taken as an AFG equation

$$\vdash x : c \oplus c = c \tag{11.9}$$

because with associativity and commutativity it is possible to collect equal elements of

$$c_1 \oplus \ldots \oplus c_m$$

so that they are adjacent, and then repeatedly deleting an element where adjacent ones are equal by applying (11.9).

Thus, we get a logic of context sets by letting models be maps from sets of contexts to sets of rigid first-order interpretations, taking accumulated contexts to be identical if they are equal as sets, taking $\oplus$ to be set union $\cup$, identifying a context c with the singleton set $\{c\}$, taking this form of Enter/Exit:

$$\text{Enter: } \frac{\vdash x : ist(c, \lambda)}{\vdash x \cup \{c\} : \lambda} \qquad \text{Exit: } \frac{\vdash x \cup \{c\} : \lambda}{\vdash x : ist(c, \lambda)} \qquad \text{where } x \in 2^C, c \in C.$$

and adding the following axioms to those common to all AFG systems:

$$\begin{array}{ll} \text{COMM:} & \vdash x : ist(c, d, \lambda) \leftrightarrow ist(d, c, \lambda) \\ \text{IDEM:} & \vdash x : ist(c, c, \lambda) \leftrightarrow ist(c, \lambda) \end{array}$$

11.4 Flat contexts

For applications where the importance of each item of context is considered to be independent of where it is examined from, the authors of [5] propose a model structure for propositional formulas asserted in context, where only the last context entered into has any bearing on validity. In [4] the same motivation and a similar constraint is applied to a quantificational logic of context. This type of context model has come to be known as 'flat'.

The following algebraic equation corresponds to flatness:

$$\vdash x : u \oplus v = v. \tag{11.10}$$

Let us reflect on some of its consequences. First, we note that (11.10) subsumes associativity:

$$(u \oplus v) \oplus w = v \oplus w = u \oplus (v \oplus w).$$

With associativity, equation (11.10) can adequately be taken as an AFG equation

$$\vdash x : c \oplus d = d \tag{11.11}$$

because given a context term containing two adjacent but otherwise arbitrary subterms $c_1 \oplus \ldots \oplus c_m$ and $d_1 \oplus \ldots \oplus d_n$:

$$b_1 \oplus \ldots \oplus b_i \oplus c_1 \oplus \ldots \oplus c_m \oplus d_1 \oplus \ldots \oplus d_n \oplus e_1 \oplus \ldots \oplus e_j,$$

the subterm $c_1 \oplus \ldots \oplus c_m$ can be removed by repeatedly removing the left neighbour of d_1 by application of (11.11).

The equivalence classes imposed by equation (11.10) are singleton sets, since associativity of $\oplus$ is implied, and each finite but nonempty sequence is equal to the singleton sequence containing its rightmost element. The equivalence classes can therefore be identified with individual contexts.

Equation (11.10) also implies that every context c is a continuation of every other context d, so any flat c-model is also a flat d-model. We can therefore talk about flat models per se, without rooting them at context terms.

This has the interesting consequence that

$$\models c : \neg ist(d, \bot)$$

in the class of all flat models. By completeness, it follows that the logic of flat contexts has the theorem

$$\vdash c : \neg ist(d, \bot)$$

which is comparable to axiom schema (D) $\neg\Box\bot$ of modal logic. Suppose in fact that a flat model M has $M(c_1) = \emptyset$ for some context c_1. Then for every rigid interpretation $I \in M(c_1)$ we have, vacuously, $M, I \models c_1 : ist(c_2, \bot)$ for arbitrary c_2, which by flatness and the model conditions is equivalent to $M, J \models c_2 : \bot$ for every $J \in M(c_2)$. This can only be the case if also $M(c_2) = \emptyset$, and then $M(c) = \emptyset$ for every c. Hence the only flat model which verifies $c : ist(d, \bot)$ is the inconsistent model, and that verifies everything, also $c : \neg ist(d, \bot)$.

The axiom schema corresponding to (11.10) is $\vdash x : ist(c, d, \lambda) \leftrightarrow ist(d, \lambda)$, but considering that x in this case will reduce to an atomic context, the following adheres to our notational conventions:

$$\text{FLAT:} \quad \vdash b : ist(c, d, \lambda) \leftrightarrow ist(d, \lambda). \tag{11.12}$$

Thus, we get a quantificational logic of flat contexts by letting models be functions from contexts to sets of rigid first-order interpretations, considering sequences of contexts to be identical according to equality of their rightmost elements, taking this form of the Enter/Exit rules:

$$\text{Enter: } \frac{\vdash b : ist(c, \lambda)}{\vdash c : \lambda} \qquad \text{Exit: } \frac{\vdash c : \lambda}{\vdash b : ist(c, \lambda)} \qquad \text{where } b, c \in C.$$

and adding axiom (11.12) to the common AFG system.

This model structure and Hilbert system for flat contexts match those in [4], but here we have arrived at them by specialization within a more general logical framework. Flat contexts were also obtained as a special case of a fibred logic of context in [8].

12 Concluding remarks

We have developed a logic of contexts which allows equational specification of context combination within the class of Associative Finite Ground algebras, and displayed several special cases including novel systems as well as systems coinciding with those of other workers in the field.

Ground equations in semigroups were first studied by Axel Thue [20] early in the 20^{th} century, and this became one of the roots of general term rewriting systems. In the literature, there is little information about non-ground algebras of context with efficiently computable normal forms. It seems that labelled deductive systems in the style of [7] would be a natural setting in which to study this topic.

The *ist* modality is by no means the only formalization of context, see [17, 2, 18, 19] for some alternatives, and e.g. [1, 3] for surveys. Some of the alternatives cater more naturally for variations in language among contexts, and multilanguage systems for context [18] seem particularly well suited to this. Multilanguage systems with hiearchical structure have been described, but apparently there are no studies so far of multilanguage systems of context with an algebraic structure of the same scope as the one we have treated here. This therefore remains an interesting topic for future study.

13 Acknowledgement

The author is grateful to an anonymous referee, who found a flaw in an earlier version of the proof of completeness, and pointed out a way to do it correctly.

References

[1] V. Akman and M. Surav. Steps toward formalizing context. *AI Magazine*, 17(3):55–72, 1996.

[2] G. Attardi and M. Simi. A formalization of viewpoints. *Fundamenta Informaticae*, 23(2-4):149–174, 1995.

[3] M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual reasoning distilled. *Journal of Theoretical and Experimental Artificial Intelligence*, 2000.

[4] Saša Buvač. Quantificational logic of context. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996.

[5] Saša Buvač, Vanja Buvač, and Ian A. Mason. Metamathematics of context. *Fundamenta Informaticae*, 23(3), 1995.

[6] Brian F. Chellas. *Modal Logic: an introduction.* Cambridge University Press, 1980.

[7] Dov Gabbay. *Labelled Deductive Systems.* Oxford Logic Guides 33; 497pp. Clarendon Press, Oxford, 1996.

[8] Dov Gabbay and Rolf Nossum. Structured contexts with fibred semantics. In P. Bonzon, M. Cavalcanti, and R. Nossum, editors, *Formal Aspects of Context.* Applied Logic Series, Kluwer, 2000.

[9] Dov M. Gabbay, Rolf Nossum, and John Woods. A contextual approach to the logic of fiction, forthcoming, 2003.

[10] F. Giunchiglia and P. Bouquet. *Introduction to Contextual Reasoning, An Artificial Intelligence Perspective*, volume 3 of *NBU Series in Cognitive Science.* Sofia, Bulgaria, May 1997.

[11] Fausto Giunchiglia. Contextual reasoning. *Epistemologia*, special issue on I Linguaggi e le Macchine(XVI):345–364, 1993. IRST Technical Report 9211-20, IRST, Trento, Italy.

[12] Ramanathan V. Guha. *Contexts: A Formalization and Some Applications.* PhD thesis, Stanford University, 1991.

[13] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

[14] Ruth Kempson, Wilfried Meyer-Viol, and Dov Gabbay. *Dynamic Syntax: The Flow of Language Understanding.* Blackwell, 2001.

[15] J. McCarthy and S. Buvač. Formalizing context, (expanded notes). In A. Aliseda, R. van Glabbeek, and D. Westerståhl, editors, *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. Stanford University, Center for the Study of Language and Information, 1998.

[16] John McCarthy. Generality in artificial intelligence. *Comm. of ACM*, 30(12):1030–1035, 1987.

[17] Rolf Nossum. A decidable multi-modal logic of context. HiA Skriftserie nr 28, Agder University College, 1997. AAAI Fall Symposium, Boston, November 1997.

[18] Luciano Serafini and Fausto Giunchiglia. ML systems: A proof theory for contexts. *Journal of Logic, Language and Information*, July 2001.

[19] R. Thomason. Type theoretic foundation for context, part 1: Contexts as complex type-theoretic objects. In P. Bouquet, L. Serafini, P. Brezillon, M. Benerecetti, and F. Castellani, editors, *Modeling and using context - proceedings of the 2nd international and interdisciplinary conference*, volume 1688 of *Lecture Notes in Artificial Intelligence*, pages 351–360. Springer Verlag, Heidelberg, 1999. 9-11 September 1999, Trento, Italy.

[20] Axel Thue. Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln. *Kristiania Videnskabsakademis Skrifter, Matematisk-Naturvidenskabelig Klasse*, 34(10), 1914.

[21] G. Wiederhold. An algebra for ontology composition. Proceedings of 1994 Monterey Workshop on Formal Methods, Sept 1994, U.S. Naval Postgraduate School, Monterey CA, pages 56-61.

[22] Martin Wirsing. *Algebraic Specification.* In: Jan van Leeuwen (ed.): Handbook of Theoretical Computer Science, volume B, chapter 13, pp 675-788. Elsevier Science Publishers, 1990.

Appendix: proofs of derived axiom schemata

The proofs of K^n, $G^{n+1,0,n,1}$, and AND^n are by induction on n. The bases, where $n = 1$, are just K, and $G^{2,0,1,1}$ in the first two cases, and for AND^n it is this:

AND
1 $\vdash x \oplus c : \lambda \wedge \chi \to \lambda$ PL
2 $\vdash x : ist(c, \lambda \wedge \chi) \to ist(c, \lambda)$ 1, K, MP
3 $\vdash x : ist(c, \lambda \wedge \chi) \to ist(c, \chi)$ symmetric
4 $\vdash x : ist(c, \lambda \wedge \chi) \to ist(c, \lambda) \wedge ist(c, \chi)$ 2, 3, PL
5 $\vdash x \oplus c : \lambda \to (\chi \to \lambda \wedge \chi)$ PL
6 $\vdash x : ist(c, \lambda) \to (ist(c, \chi) \to ist(c, \lambda \wedge \chi))$ 5, K twice, PL
7 $\vdash x : ist(c, \lambda) \wedge ist(c, \chi) \to ist(c, \lambda \wedge \chi)$ 6, PL
8 $\vdash x : ist(c, \lambda \wedge \chi) \leftrightarrow ist(c, \lambda) \wedge ist(c, \chi)$ 4,7, PL

For the induction steps, fix an $n > 0$ and assume K^n, $G^{n+1,0,n,1}$, and AND^n as inductive hypotheses.

K^{n+1}
1 $\vdash x \oplus c_1 : ist(c_2, \ldots, c_{n+1}, \lambda \to \chi) \to (ist(c_2, \ldots, c_{n+1}, \lambda) \to ist(c_2, \ldots, c_{n+1}, \chi))$ ind. hyp.
2 $\vdash x : ist(c_1, ist(c_2, \ldots, c_{n+1}, \lambda \to \chi) \to (ist(c_2, \ldots, c_{n+1}, \lambda) \to ist(c_2, \ldots, c_{n+1}, \chi)))$ 1, Exit
3 $\vdash x : ist(c_1, \ldots, c_{n+1}, \lambda \to \chi) \to (ist(c_1, \ldots, c_{n+1}, \lambda) \to ist(c_1, \ldots, c_{n+1}, \chi))$ 2, K twice

$G^{n+2,0,n+1,1}$
1 $\vdash x \oplus c_1 : \neg ist(c_2, \ldots, c_{n+2}, \lambda) \to ist(c_2, \ldots, c_{n+1}, \neg ist(c_{n+2}, \lambda))$ ind. hyp.
2 $\vdash x : ist(c_1, \neg ist(c_2, \ldots, c_{n+2}, \lambda) \to ist(c_2, \ldots, c_{n+1}, \neg ist(c_{n+2}, \lambda)))$ 1, Exit
3 $\vdash x : \neg ist(c_1, \ldots, c_{n+2}, \lambda) \to ist(c_1, \ldots, c_{n+1}, \neg ist(c_{n+2}, \lambda))$ 2, $G^{2,0,1,1}$, PL

AND^{n+1}
1 $\vdash x \oplus c_1 : ist(c_2, \ldots, c_{n+1}, \lambda \wedge \chi) \leftrightarrow ist(c_2, \ldots, c_{n+1}, \lambda) \wedge ist(c_2, \ldots, c_{n+1}, \chi)$ ind. hyp.
2 $\vdash x : ist(c_1, ist(c_2, \ldots, c_{n+1}, \lambda \wedge \chi) \leftrightarrow ist(c_2, \ldots, c_{n+1}, \lambda) \wedge ist(c_2, \ldots, c_{n+1}, \chi))$ 1, Exit
3 $\vdash x : ist(c_1, \ldots, c_{n+1}, \lambda \wedge \chi) \leftrightarrow ist(c_1, ist(c_2, \ldots, c_{n+1}, \lambda) \wedge ist(c_2, \ldots, c_{n+1}, \chi)))$ 2, K twice
4 $\vdash x : ist(c_1, \ldots, c_{n+1}, \lambda \wedge \chi) \leftrightarrow ist(c_1, \ldots, c_{n+1}, \lambda) \wedge ist(c_1, \ldots, c_{n+1}, \chi)$ 3, AND, PL

This completes the inductive proofs of K^n, $G^{n+1,0,n,1}$, and AND^n. The schema OR^n is an easy consequence of K^n :

OR^n
1 $\vdash x : \lambda \to \lambda \vee \chi$ PL
2 $\vdash x : ist(c_1, \ldots, c_n, \lambda) \to ist(c_1, \ldots, c_n, \lambda \vee \chi)$ 1, K^n
3 $\vdash x : ist(c_1, \ldots, c_n, \chi) \to ist(c_1, \ldots, c_n, \lambda \vee \chi)$ symmetric
4 $\vdash x : ist(c_1, \ldots, c_n, \lambda) \vee ist(c_1, \ldots, c_n, \chi) \to ist(c_1, \ldots, c_n, \lambda \vee \chi)$ 2, 3, PL

The proof of $\Delta^{\pm}$ is in two parts, depending on the sign of $\pm ist(d, \lambda)$. First, the case where the sign is positive:

Δ^{+}

1 $\vdash x : \neg ist(c, ist(d, \lambda)) \rightarrow ist(c, \neg ist(d, \lambda))$ $\quad G^{2,0,1,1}$

2 $\vdash x : (ist(c, \neg ist(d, \lambda)) \rightarrow ist(c, \chi)) \rightarrow (\neg ist(c, ist(d, \lambda)) \rightarrow ist(c, \chi))$ $\quad$ 1, PL

3 $\vdash x : ist(c, \neg ist(d, \lambda) \rightarrow \chi) \rightarrow (ist(c, \neg ist(d, \lambda)) \rightarrow ist(c, \chi))$ $\quad$ K

4 $\vdash x : ist(c, \neg ist(d, \lambda) \rightarrow \chi) \rightarrow (\neg ist(c, ist(d, \lambda)) \rightarrow ist(c, \chi))$ $\quad$ 2, 3, PL

5 $\vdash x : ist(c, ist(d, \lambda) \vee \chi) \rightarrow ist(c, ist(d, \lambda)) \vee ist(c, \chi)$ $\quad$ 4, PL

Finally, the case where the sign is negative:

Δ^-

1 $\vdash x : \neg ist(c, \neg ist(d, \lambda)) \rightarrow ist(c, ist(d, \lambda))$ $\quad G^{2,0,1,1}, PL$

2 $\vdash x : (ist(c, ist(d, \lambda)) \rightarrow ist(c, \chi)) \rightarrow (\neg ist(c, \neg ist(d, \lambda)) \rightarrow ist(c, \chi))$ $\quad$ 1, PL

3 $\vdash x : ist(c, ist(d, \lambda) \rightarrow \chi) \rightarrow (ist(c, ist(d, \lambda)) \rightarrow ist(c, \chi))$ $\quad$ K

4 $\vdash x : ist(c, ist(d, \lambda) \rightarrow \chi) \rightarrow (\neg ist(c, \neg ist(d, \lambda)) \rightarrow ist(c, \chi))$ $\quad$ 2, 3, PL

5 $\vdash x : ist(c, \neg ist(d, \lambda) \vee \chi) \rightarrow ist(c, \neg ist(d, \lambda)) \vee ist(c, \chi)$ $\quad$ 4, PL

∎

Received October 22, 2001