George Tourlakis

A BASIC FORMAL EQUATIONAL PREDICATE LOGIC – PART II

Abstract
We continue our exploration of the “Basic Formal Equational Predicate Logic” of Part I. Section numbering is consecutive to that of Part I. We show that a strong “no-capture” Leibniz, and a weak “full-capture” version are derived rules (both access the interior of quantifier scopes). We derive general rules MON (monotonicity) and AMON (antimonotonicity) that are “as strong as possible” for our logic. Finally, we show that our logic is sound and complete. The bibliography at the end applies to Part II only.

4. Derived Leibniz rules
In this section we introduce some derived rules of the type “Leibniz”. Note that we do not have the following Strong Leibniz with Uniform Substitution.

\[ A \equiv B \]
\[ C[p \setminus A] \equiv C[p \setminus B] \]  

That SLUS is “invalid” in our Logic follows from a proof in [9] that SLUS implies “strong generalization” (take \( B = \text{true} \), \( C = (\forall x)p \)).

**Metatheorem 4.1. [Strong Leibniz with Contextual Substitution—SLCS]** The following is a derived rule:

\[ A \equiv B \]
\[ C[p := A] \equiv C[p := B] \]  

**SLCS**
Proof. This was proved in [9] by induction on the formula C. Only the induction step when \( C = \forall x D \) is interesting. By I.H., \( A \equiv B \vdash D[p := A] \equiv D[p := B] \), hence (by the Tautology theorem) \( A \equiv B \vdash (\forall x) D[p := A] \supset D[p := B] \). Since \( x \) is not free in \( A \equiv B \), 3.11 (Part I) yields \( A \equiv B \vdash (\forall x) D[p := A] \supset (\forall x) D[p := B] \).

Similarly (from \( A \equiv B \vdash D[p := A] \iff D[p := B] \)), \( A \equiv B \vdash (\forall x) D[p := A] \iff (\forall x) D[p := B] \), hence, one more application of the Tautology theorem gives \( A \equiv B \vdash (\forall x) D[p := A] \equiv (\forall x) D[p := B] \).

\[ \square \]

Metatheorem 4.2. [Weak Leibniz with Uniform Substitution—WLUS] The following is a derived rule: If \( \vdash A \equiv B \) then \( \vdash C[p \setminus A] \equiv C[p \setminus B] \).

Proof. The proof is as above, with the following differences: We assume \( \vdash A \equiv B \) throughout. The induction step where \( C = (\forall x) D \) now is: By I.H. we have \( \vdash (\forall x) D[p \setminus A] \equiv (\forall x) D[p \setminus B] \). By the Tautology theorem \( \vdash (\forall x) D[p \setminus A] \supset (\forall x) D[p \setminus B] \), thus \( \vdash (\forall x) D[p \setminus A] \iff (\forall x) D[p \setminus B] \) by 3.11 (Part I). Similarly we obtain \( \vdash (\forall x) D[p \setminus A] \iff (\forall x) D[p \setminus B] \) and are done by the Tautology theorem. \[ \square \]

5. Monotonicity

We address in this section a “weakness” of the current literature ([1], [4]) on equational or calculational reasoning – to which David Gries has already called attention in [3]. That is, while it is customary to mix “\( \equiv \)”-steps (applications of a conjunctional “\( \equiv \)” and “\( \Rightarrow \)”-steps (applications of a conjunctional “\( \Rightarrow \)” in a calculational proof, and while there are well documented rules to handle the former,\(^1\) yet the latter type of step\(^2\) seems to usually rely on a compendium of ad hoc rules. We hope to have contributed towards remedying this state of affairs by presenting here a unified yet simple and rigorous way to understand, ascertain validity, and

\[ \quad \]
\[ ^1 \text{That is, to “calculate”, and so annotate, a new equivalence from a given one —via Leibniz.} \]
\[ ^2 \text{That is, to “calculate”, and appropriately annotate, a new implication from a given one.} \]
therefore annotate and utilize \( \Rightarrow \)-steps, using the rules monotonicity and antimonotonicity.

\textbf{Definition 5.1.} We define a set of strings, the \textit{I-Forms} and \textit{D-Forms}, by induction. It is the smallest set of strings over the alphabet \( V \cup \{\ast\} \) —where \( \ast \) is a new symbol added to the alphabet \( V \) (Part I, Section 1)— satisfying the following:

\textbf{Fm1.} \( \ast \) is an I-Form (but not a D-form).

\textbf{Fm2.} If \( A \) is any formula and \( \mathcal{U} \) is an I-Form (respectively, D-Form), then the following are also I-Forms (respectively, D-Forms): \((\mathcal{U} \lor A)\), \((A \lor \mathcal{U})\), \((A \land \mathcal{U})\), \((\mathcal{U} \Rightarrow A)\), \(((\forall x)\mathcal{U})\) and \(((\exists x)\mathcal{U})\),\(^3\) while the following are D-Forms (respectively, I-Forms): \((\neg \mathcal{U})\) and \((\mathcal{U} \Rightarrow A)\).

We will just say \( \mathcal{U} \) is a Form, if we do not wish to spell out its type (I or D). We will use calligraphic capital letters \( \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y} \) to denote Forms.

\textbf{Definition 5.2.} For any Form \( \mathcal{U} \) and any formula \( A \) or form \( W \), the symbols \( \mathcal{U}[A] \) and \( \mathcal{U}[W] \) mean, respectively, the result of the uniform substitutions \( \mathcal{U}[\ast \ A] \) and \( \mathcal{U}[\ast \ W] \).

Our I-Forms and D-Forms ("I" for \textit{increasing}, and "D" for \textit{decreasing}) are motivated by, but are different from, the Positive and Negative Forms of Schütte [7].\(^4\) The expected behaviour of the Forms is that they are "monotonic functions" of the \( \ast \)-"variable" in the following sense: We expect that \( \vdash A \Rightarrow B \) will imply \( \vdash \mathcal{U}[A] \Rightarrow \mathcal{U}[B] \) if \( \mathcal{U} \) is an I-Form, and \( \vdash \mathcal{U}[A] \Leftarrow \mathcal{U}[B] \) if it is a D-Form.

\textbf{Lemma 5.3.} Every Form contains exactly one occurrence of \( \ast \) as a substring.

\textbf{Proof.} Induction on Forms. The basis is immediate. Moreover, the property we are asked to prove obviously "propagates" with the formation rules. \hfill \Box

\textbf{Lemma 5.4.} For any Form \( \mathcal{U} \) and formula \( A \), \( \mathcal{U}[A] \) is a formula.

\( ^3 \)Even though "\( \exists \)" is a metasymbol (Section 1, Part I), it is used widely enough to warrant mapping out its behaviour explicitly.

\( ^4 \)For example, \((\ast \land A)\) is an I-Form but not a Positive Form in the sense of [7], since the latter would necessitate, in particular, that \((\text{true} \land A)\) be a tautology.
Proof. Induction on Forms.

Lemma 5.5. No Form has (is of) both types I and D.

Proof. Let \( \mathcal{U} \) have the least complexity among forms that have both types. This is not the “basic” Form \( \ast \) as that is declared to have just type I. Can it be a form \( (V \vee A) \)? No, for if it has both types I and D, \( V \) must also have both types I and D, contradicting the assumption that \( \mathcal{U} \) was the least complex “schizophrenic” Form. We obtain similar contradictions in the case of all the other formation rules.

Lemma 5.6. For any Forms \( \mathcal{U} \) and \( \mathcal{V} \), we have the following composition properties:

1. If \( \mathcal{U} \) is an I-Form, then \( \mathcal{U}[\mathcal{V}] \) has the type of \( \mathcal{V} \).
2. If \( \mathcal{U} \) is an D-Form, then \( \mathcal{U}[\mathcal{V}] \) has the type opposite to that of \( \mathcal{V} \).

Proof. We do induction on \( \mathcal{U} \) to prove (1) and (2) simultaneously. The basis is obvious, as \( \mathcal{U} = \ast \), hence \( \mathcal{U}[\mathcal{V}] = \mathcal{V} \).

Case 1. \( \mathcal{U} = (\mathcal{W} \vee A) \), for some \( A \in \text{Wff} \). \( \mathcal{U}[\mathcal{V}] = (\mathcal{W}[\mathcal{V}] \vee A) \). \( \mathcal{U} \) and \( \mathcal{W} \) have the same type. By I.H. and the definition of Forms the claim follows. We omit a few similar cases . . .

Case 2. \( \mathcal{U} = (\mathcal{W} \Rightarrow A) \), for some \( A \in \text{Wff} \). \( \mathcal{U}[\mathcal{V}] = (\mathcal{W}[\mathcal{V}] \Rightarrow A) \). \( \mathcal{U} \) and \( \mathcal{W} \) have opposite types. By I.H. and the definition of Forms the claim follows. We omit a few similar cases . . .

Case 3. \( \mathcal{U} = ((\forall x)\mathcal{W}) \). \( \mathcal{U}[\mathcal{V}] = ((\forall x)\mathcal{W}[\mathcal{V}]) \). \( \mathcal{U} \) and \( \mathcal{W} \) have the same type. By I.H. and the definition of Forms the claim follows. We omit the “∃-case”.

Remark 5.7. Thus, if \( \mathcal{U} \) is obtained by a chain of compositions, it is an I-Form if the chain contains an even number of D-Forms, it will be a D-Form otherwise. For example, if \( \mathcal{U} \) is a D-Form, then \( \mathcal{U}[\mathcal{A} \Rightarrow \ast] \) is still a D-Form, but \( \mathcal{U}[\ast \Rightarrow \mathcal{A}] \) is an I-Form.

Metatheorem 5.8. [Monotonicity and Antimonotonicity] Let \( \vdash A \Rightarrow B \). If \( \mathcal{U} \) is an I-Form, then \( \vdash \mathcal{U}[\mathcal{A}] \Rightarrow \mathcal{U}[\mathcal{B}] \), else (a D-Form) \( \vdash \mathcal{U}[\mathcal{A}] \Leftarrow \mathcal{U}[\mathcal{B}] \).

NB. We call MON the rule “if \( \vdash A \Rightarrow B \) and \( \mathcal{U} \) is an I-Form, then \( \vdash \mathcal{U}[\mathcal{A}] \Rightarrow \mathcal{U}[\mathcal{B}] \)”. We call AMON the rule “if \( \vdash A \Rightarrow B \) and \( \mathcal{U} \) is an D-Form, then \( \vdash \mathcal{U}[\mathcal{A}] \Leftarrow \mathcal{U}[\mathcal{B}] \)".
Proof. Induction on $U$.

Basis. $U = \ast$, hence we want to prove $\vdash A \Rightarrow B$, which is the same as the hypothesis. The induction steps:

**Case 1.** $U = (W \lor C)$, for some $C \in \text{Wff}$. If $W$ is an I-Form, then (I.H.) $\vdash W[A] \Rightarrow W[B]$, hence $\vdash (W[A] \lor C) \Rightarrow (W[B] \lor C)$ by tautological implication. If $W$ is a D-Form, then (I.H.) $\vdash W[A] \Leftarrow W[B]$, hence $\vdash (W[A] \lor C) \Leftarrow (W[B] \lor C)$ by tautological implication.

**Case 2.** $U = (W \Rightarrow C)$, for some $C \in \text{Wff}$. If $W$ is an I-Form, then (I.H.) $\vdash W[A] \Rightarrow W[B]$, hence $\vdash (W[A] \Rightarrow C) \Rightarrow (W[B] \Rightarrow C)$ by tautological implication. If $W$ is a D-Form, then (I.H.) $\vdash W[A] \Leftarrow W[B]$, hence $\vdash (W[A] \Rightarrow C) \Leftarrow (W[B] \Rightarrow C)$ by tautological implication. We omit a few similar cases based on tautological implication . . .

**Case 3.** $U = (\forall x)W$. If $W$ is an I-Form, then (I.H.) $\vdash W[A] \Rightarrow W[B]$. By 3.11 (Part I), $\vdash (\forall x)W[A] \Rightarrow (\forall x)W[B]$. If $W$ is a D-Form, then (I.H.) $\vdash W[A] \Leftarrow W[B]$. By 3.11 (Part I), $\vdash (\forall x)W[A] \Leftarrow (\forall x)W[B]$.

**Case 4.** $U = (\exists x)W$. As above, but relying on 3.12 (Part I) instead.

$\Box$

NB. MON and AMON are applied after we have eliminated the presence of $\equiv$ from formulas. If we are willing to weaken the type of substitution we effect into Forms, we can strengthen the type of premise in MON and AMON:

**Metatheorem 5.9.** [“Strong” MON/AMON with contextual substitution] The following are derived rules:

- $A \Rightarrow B \vdash U[* := A] \Rightarrow U[* := B]$, if $U$ is an I-Form
- $A \Rightarrow B \vdash U[* := A] \Leftarrow U[* := B]$, if $U$ is a D-Form

Proof. The proof is as in 5.8, except that the induction steps under cases 3 and 4 are modified as follows:

**Case 3.** $U = (\forall x)W$. If $W$ is an I-Form, then (I.H.)

$$A \Rightarrow B \vdash W[* := A] \Rightarrow W[* := B] \quad (i)$$

We want to argue that

$$A \Rightarrow B \vdash (\forall x)W[* := A] \Rightarrow (\forall x)W[* := B] \quad (ii)$$
where we have already incorporated “((∀x)W)[* := A] = (∀x)(W[* := A])”, etc., and then dropped the unnecessary brackets. Now, if the substitutions in (ii) are not defined, then there is nothing to state (let alone prove). Assuming that they are defined, then $A \Rightarrow B$ has no free occurrence of $x$. By 3.11 (Part I), (ii) follows from (i). If $W$ is a D-Form, then we argue as above on the I.H.

$$A \Rightarrow B \vdash W[* := A] \iff W[* := B].$$

**Case 4.** $U = (∃x)W$. We use here 3.12 (Part I) instead of 3.11 (Part I).

**Remark 5.10.** The above is as far as it goes. If we allow uniform substitution as well, to obtain “Extra strong MON/AMON”, then the rule yields strong generalization, hence it is not valid in E-logic. The following calculation illustrates this point.

$$A$$

\[
= \{ \models \text{Taut} \ A \equiv true \Rightarrow A \}
\]

\[
\Rightarrow \text{ (“Extra strong MON” and the I-Form (∀x)*)}
\]

\[
(\forall x)true \Rightarrow (\forall x)A
\]

\[
= (\text{PSL and } \vdash (\forall x)true \Rightarrow true (\text{Ax2}), \vdash true \Rightarrow (\forall x)true (\text{Ax3}))
\]

\[
\Rightarrow true \Rightarrow (\forall x)A
\]

\[
= \{ \models \text{Taut} \ A \equiv true \Rightarrow A \}
\]

\[
\Rightarrow (\forall x)A
\]

The above ⇒ (on the left margin) is, of course, ⇒ (no difference in “power”, by the Deduction Theorem). Thus we have just “proved” $A \vdash (∀x)A$.

### 6. Soundness and Completeness of E-logic

The semantics must accurately reflect our syntactic choices – most important of which was to disallow strong generalization. In particular, we will define “logically implies”, $\models$, so that “$A \models B$ iff $\models A \Rightarrow B$” holds, in order to reflect accurately the fact that “$A \vdash B$ iff $\vdash A \Rightarrow B$.”
Definition 6.1. Given a language $L = (V, \text{Term}, \text{Wff})$, a structure $M = (M, I)$ appropriate for $L$ is such that $M \neq \emptyset$ is a set (the “domain”) and $I$ is an “interpretation” mapping that assigns

1. to each constant $a$ of $V$ a unique member $a^I \in M$
2. to each function $f$ of $V$ of arity $n$ a unique (total) function $f^I : M^n \to M$
3. to each predicate $P$ of $V$ of arity $n$ a unique set $P^I \subseteq M^n$
4. to each propositional variable $p$ of $V$ a unique member of the two element set $\{t, f\}$ (we understand $t$ as “true” and $f$ as “false”)

Moreover we set $\text{true}^I = t$ and $\text{false}^I = f$, where the use of “=” here is metamathematical (equality on $\{t, f\}$).

Remark 6.2. $I$ is the “primary map” that gives meaning to all basic symbols. To extend the meaning to terms and formulas one introduces a “state”, $\sigma$, for object variables, that is, a mapping $\sigma : \text{object variables} \to M$ that in effect assigns fixed values to them ($\sigma(x_i) \in M$, for $i \geq 1$). One then defines the “value” of any term $t$ and any formula $A$ “at $\sigma$” by induction on terms and formulas respectively. These values are denoted by “$t[\sigma]$” (in $M$) and “$A[\sigma]$” (in $\{t, f\}$) respectively. We omit the standard inductive definition (it is correctly presented in, for example, [2], [5] and is trivially adaptable to allow 6.1(4,5)).

Definition 6.3. Let $A \in \text{Wff}$ and $M$ be a structure as above. We say that $A$ is satisfiable in $M$ with state $\sigma$, in symbols $\models_M A[\sigma]$, iff $A[\sigma] = t$ in $M$. We say that $M$ is a model of $A$, in symbols $\models_M A$—note the absence of $\sigma$—iff $(\forall \sigma) \models_M A[\sigma]$. For any set of formulas $\Gamma$ from $\text{Wff}$, $\models_M \Gamma$ denotes the sentence “$M$ is a model of $\Gamma$”, and means that for all $A \in \Gamma$, $\models_M A$.

A formula $A$ is universally valid (we often say just valid) iff every structure appropriate for the language is a model of $A$. Under these circumstances we simply write $\models A$.

Definition 6.4. [Crucial] We say that $\Gamma$ logically implies $A$, in symbols $\Gamma \models A$, to mean that

$$(\forall M, \forall \sigma) \left( (\forall B \in \Gamma) \left( \models_M B[\sigma] \right) \imply \models_M A[\sigma] \right)$$

In most of the literature, parts 6.1(4,5) are absent. [7] however does employ Boolean variables and constants.
where (informal) quantification $\forall M$ is over all appropriate structures for the language of $\Gamma \cup \{A\}$.

Remarks 6.5.

(1) Clearly, in the case of $A \models B$ the above says that, having fixed the structure $M$, every state $\sigma$ that makes $A$ true makes $B$ true. Thus, $A \models B$ exactly when $\models A \Rightarrow B$, as earlier promised.

(2) Approaches that allow strong generalization must use a different definition of "$\Gamma \models A$". This is done correctly in [8], [9] as,

$$(\forall M)\left( (\forall B \in \Gamma) \left( \models_M B \right) \text{ implies } \models_M A \right)$$

Definition 6.6. [First order theories] A (first order) theory is a collection of the following objects: A first order language $L = (V, \text{Term}, \text{Wff})$; a set of logical axioms; a set of rules of inference; a set of nonlogical axioms, plus a definition of "deduction" (proof) and "theorem" (2.12 and 2.10, Part I).

Remark 6.7. How is this different from a first order "Logic"? The latter terminology emphasizes our interest in the rules of inference and logical axioms and our disinterest in any specific nonlogical symbols and nonlogical axioms—we discuss these "in general". A specific theory on the other hand is specifically designed to produce the consequences of specific nonlogical axioms (e.g., set theory).

We often name a theory by the name of its nonlogical axioms (as in "let $\Gamma$ be a theory . . ."), but we may also name a theory by other characteristics, e.g., the choice of language, rules of inference, etc. For example, that $A$ is provable from $\Gamma$ within E-logic (Part I, p.9) may be denoted by $\Gamma \vdash E A$ while provability from the same nonlogical axioms within En-logic (Part I, p.9) would be denoted by $\Gamma \vdash En A$. An E-theory (En-theory) is developed within E-logic (En-logic). A pure theory is one with $\Gamma = \emptyset$.

Definition 6.8. [Soundness] A theory $\Gamma$ is sound, iff $\Gamma \vdash A$ implies $\models A$.

Metatheorem 6.9. Any first order E-theory is sound.

---

6Incidentally, the logic in [6] does support strong generalization, yet the wrong semantics (namely, 6.4) are given.
Proof. The proof — working with an En-theory due to (3.5; Part I)— is (essentially) contained in [2] and is omitted. The fact that formulas in [2] contain neither Boolean variables nor constants does not detract from the straightforward proof principle: Induction on Γ-theorems using the fact that modus ponens preserves truth, and that the logical axioms are universally valid. \[\square\]

NB. A by-product of soundness is consistency. An (E-)theory Γ is consistent iff \(\text{Thm}_\Gamma \subseteq \text{Wff}\) (proper subset). Thus, any pure E-theory is consistent, since, by soundness, \(\not\vdash \text{false}\).

Definition 6.10. A theory Γ is complete iff \(\Gamma \models A\) implies \(\Gamma \vdash A\) for any formula A.

We will show the completeness of any pure E-theory\(^7\) by proving that it extends conservatively\(^8\) the E-theory obtained by leaving all else the same but dropping all Boolean variables and constants from the alphabet \(V\). (The latter theory is shown to be complete in, e.g., [2].)

We employ two technical lemmata:

Lemma 6.11. [Substitution into propositional variables] In any E-theory (En-theory) Γ, if \(\Gamma \vdash A\), with a condition on the proof, then \(\Gamma \vdash A[p:=W]\) for any formula W and propositional variable p. The condition is: The propositional variable p does not occur in any formula of Γ used in the proof of A.

Proof. Induction on the length \(n\) of the Γ-proof of A. We may assume that we are working in En-logic.

Say that a proof (satisfying the condition) is

\[B_1, \ldots, B_n\]  \hspace{1cm} (1)

where \(B_n = A\).

\(^7\)Of course, any E-theory is complete, as a reader familiar with the proof of Gödel’s Completeness theorem can directly show in 10 pages or less. Here we want to bypass a direct proof, thus restricting attention to pure theories for convenience.

\(^8\)A theory \(T'\) over the language \(L'\) is a conservative extension of a theory \(T\) over the language \(L\), if, first of all, every theorem of \(T\) is a theorem of \(T'\), and (the conservative part) moreover, any theorem of \(T'\) that is over \(L\) — the language of \(T\) — is also a theorem of \(T\). That is, \(T'\) proves no new theorems in the old language.
Basis. \( n = 1 \). Suppose that \( A \) is a logical axiom. Then \( A[p := W] \) is as well (the reader may want to refer to the axiom-list in Part I). Thus, \( \Gamma \vdash A[p := W] \). Suppose that \( A \) is a nonlogical axiom. Then \( A[p := W] = A \) by the condition on the proof, thus \( \Gamma \vdash A[p := W] \).

On the I.H. that the claim is fine for proof-lengths < \( n \), let us address the case of \( n \): If \( A \) (i.e., \( B_n \)) is logical or nonlogical, then we have nothing to add. So let \( A \sim (B_i \Rightarrow A) \) in (1) above, where \( i \) and \( j \) are each less than \( n \) (i.e., the last step of the proof was an application of modus ponens).

By I.H., \( \Gamma \vdash B_i[p := W] \) and \( \Gamma \vdash B_i[p := W] \Rightarrow A[p := W] \). Thus, by modus ponens, \( \Gamma \vdash A[p := W] \).

**Lemma 6.12.** [Main Lemma] ([9]) Let \( A \) be a formula over the language \( L \) of Part I, Section 1. Let \( p \) be a propositional variable that occurs in \( A \). Expand the language \( L \) by adding \( P \), a new 1-ary predicate symbol.

Then, \( \models A \) iff \( \models A[p := (\forall x)Px] \) and \( \vdash A \) iff \( \vdash A[p := (\forall x)Px] \).

**Proof.** The proof is contained in [9] and is therefore omitted. In each case (\( \vdash \), and \( \models \)) the only-if direction employs Lemma 6.11.

**Corollary 6.13.** ([9]) Any pure E-theory is complete.

**Proof.** Fix attention to the pure E-theory over a fixed language \( L \). Let \( A \) be a formula in the language, and let \( \models A \). Denote by \( A' \) the formula obtained from \( A \) by replacing each occurrence of \( \text{true} \) (respectively \( \text{false} \)) by \( p \lor \neg p \) (respectively \( p \land \neg p \)) where \( p \) is a propositional variable not occurring in \( A \). Let \( A'' \) be obtained from \( A' \) by replacing each propositional variable \( p, q, \ldots \) in it by \( (\forall x)Px \), \((\forall x)Qx, \ldots \) respectively, where \( P, Q, \ldots \) are new predicate symbols (so we expand \( L \) by these additions).

Clearly, by 6.12, \( \models A'' \). This formula is in the language of [2]. Thus, by completeness of pure En-theories/E-theories over such a “restricted” language (proved in [2]), \( \vdash A'' \), where the proof is being carried out in the restricted language.

But, trivially, this proof is valid over the language \( L \) (same axioms, same rules), hence also \( \vdash A' \), by 6.12.

Finally, by WLUS —since \( \vdash p \lor \neg p \equiv \text{true} \) and \( \vdash p \land \neg p \equiv \text{false} \)— and EQN, we get \( \vdash A \).
7. Appendix

The reader is referred to [9] where all the axioms and rules in [4], chapters 8 and 9, were shown to be derived in the logic of [9]. Practically identical proofs are available within our present E-logic, and they will not be repeated here. We will only offer some remarks on the twin rules “Leibniz (8.12)” ([4], p. 148):

\[ A \equiv B \vdash (\forall x)(C[p := A] \Rightarrow D) \equiv (\forall x)(C[p := B] \Rightarrow D) \quad (8.12a) \]

and

\[ D \Rightarrow (A \equiv B) \vdash (\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B]) \quad (8.12b) \]

Let us prove the “weak”, “full-capture”, versions (1) and (2) below.\(^9\)

\[ \vdash A \equiv B \text{ implies } \vdash (\forall x)(C[p := A] \Rightarrow D) \equiv (\forall x)(C[p := B] \Rightarrow D) \quad (1) \]

and

\[ \vdash A \equiv B \text{ implies } \vdash (\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B]) \quad (2) \]

Now, implication (1) is an instance of WLUS, where, without loss of generality, \(p\) occurs only in \(C\). So it holds.

(2) has an identical proof. But what happened to the \(D \Rightarrow \)-part on the premise side? It was dropped, because the rule is invalid with it (see also [9]). Indeed, take \(D = x \approx 0, C = (\forall x)p, A = x \approx 0\) and \(B = true\). Then,

\[ \models D \Rightarrow (A \equiv B) \]

that is

\[ \models x \approx 0 \Rightarrow (x \approx 0 \equiv true) \]

but

\[ \not\models (\forall x)(D \Rightarrow C[p := A]) \equiv (\forall x)(D \Rightarrow C[p := B]) \]

that is

\(^9\)It is obvious that we cannot do any better: Full-capture “strong” versions will yield strong generalization.
\[ \not\vdash (\forall x)(x \approx 0 \Rightarrow (\forall x)x \approx 0) \equiv (\forall x)(x \approx 0 \Rightarrow (\forall x)true) \]

As an aside, (8.12b) is valid if the premise has an absolute proof:

\[ \vdash D \Rightarrow (A \equiv B) \]

To prove the conclusion of (8.12b), establish instead

\[ \vdash D \Rightarrow (C[p := A]) \equiv C[p := B]) \]  (3)

To this end, assume \( D \). This yields \( A \equiv B \) \(^{10}\) (by modus ponens and the premise of (8.12b)). By SLCS, \( C[p := A] \equiv C[p := B] \) follows, and hence so does (3) (Deduction Theorem).

Now, the formula in (3) yields

\[ \vdash (D \Rightarrow C[p := A]) \equiv (D \Rightarrow C[p := B]) \]  (4)

which (Tautology Theorem) yields

\[ \vdash (D \Rightarrow C[p := A]) \Rightarrow (D \Rightarrow C[p := B]) \]  (5)

and

\[ \vdash (D \Rightarrow C[p := A]) \Leftarrow (D \Rightarrow C[p := B]) \]  (6)

Since both (5) and (6) are absolute theorems, MON —on the I-Form \( (\forall x)^* \)— and the Tautology Theorem conclude the argument.

Note that we cannot do any better: If (8.12b) is taken literally (that is, “strongly”), then it yields the invalid in E-logic strong generalization \( X \vdash (\forall x)X \) (take \( D = \neg X, A = false, B = true, C = p \) in (8.12b)).

References


\(^{10}\) Not as an absolute theorem!


Acknowledgements. I wish to thank my colleague Jonathan Ostroff for the many illuminating discussions we have had about Equational Logic.

Department of Computer Science
York University
Toronto, Ontario
M3J 1P3
Canada
e-mail: gt@cs.yorku.ca