Hongkai Yin
Hanoch Ben-Yami

The Quantified Argument Calculus with Two- and Three-valued Truth-valuational Semantics

Abstract. We introduce a two-valued and a three-valued truth-valuational substitutional semantics for the Quantified Argument Calculus (Quarc). We then prove that the 2-valid arguments are identical to the 3-valid ones with strict-to-tolerant validity. Next, we introduce a Lemmon-style Natural Deduction system and prove the completeness of Quarc on both two- and three-valued versions, adapting Lindenbaum's Lemma to truth-valuational semantics. We proceed to investigate the relations of three-valued Quarc and the Predicate Calculus (PC). Adding a logical predicate T to Quarc, true of all singular arguments, allows us to represent PC quantification in Quarc and translate PC into Quarc, preserving validity. Introducing a weak existential quantifier into PC allows us to translate Quarc into PC, also preserving validity. However, unlike the translated systems, neither extended system can have a sound and complete proof system with Cut, supporting the claim that these are basically different calculi.

Keywords: Quantified argument calculus, Truth-valuational semantics, Substitutional quantification, Lindenbaum's lemma, Three-valued semantics, Strict-to-tolerant validity, Completeness.

1. Introduction

The Quantified Argument Calculus (Quarc), introduced in [2], has since been the subject of several publications, extending and applying it in a variety of ways [3,4,15,21,23–25,30], and a number of researchers are currently exploring it in additional directions. Still, a direct completeness proof for the Lemmon-style Natural Deduction system used in the original paper hasn't been published in any journal. This is one aim of the present work. Our proof, adapting Lindenbaum's construction to a new formal system with truth-valuational semantics, is of some interest in its own right. In addition, as has been discussed elsewhere [4,15], it is natural to have a three-valued version of Quarc. We shall develop such a three-valued system, proving that

Presented by **Heinrich Wansing**; *Received* May 25, 2021

https://doi.org/10.1007/s11225-022-10022-5

its consequence relation coincides with that of the two-valued system, but unlike [15], we shall do it with a truth-valuational semantics and not a model-theoretic one, and with a different conception of validity than the one used in that paper. We shall then explore the relation of that three-valued Quarc to the Predicate Calculus (PC).

Quarc was developed with the aim of being closer than PC to natural language, primarily in the way it incorporates quantification but also in other, related ways: inclusion of modes of predication, of reordered relation terms, anaphora, and more. As the calculus has by now been motivated and informally introduced in several publications, we shall do neither here, apart from concisely presenting its approach to quantification.

Consider the sentences,

1. Alice is prudent
2. Every student is prudent.

While 'Alice' occupies the subject or argument position in (1), this position is occupied by 'every student' in (2). Namely, the quantifier 'every' followed by the unary predicate 'student' form the *quantified argument* of that sentence. Quarc follows this analysis of quantification. With the argument written to the left of the predicate, it formalises these two sentences as:

3. $(a)P$
4. $(\forall S)P$.

Let us next consider particular or specific quantification. The sentence,

5. Some students are prudent

will be formalised,

6. $(\exists S)P$

where $\exists$ is read 'some', not 'there is' or 'there exists'.

It has been argued in several works [1,2,4] that particular quantification in natural language *is not related to existence* (see also [27]chap. 18), and this claim has been used to address several philosophical puzzles [4]. Although the use of an *existential* quantifier is legitimate if we are not interested in being faithful to the logic and concepts of natural language, Quarc *is* interested in the latter. We accordingly read particular quantification as having no *ontological commitment* and as unrelated to existence or to a formal existential quantifier. To emphasise this, we could follow a suggestion found in some works mentioned above and formalise it by an inverted P, from

Particular or the Latin *Particularis*, and not an inverted E, from *Existence* [26, 47]. But as this distinction is not essential for formal work in this paper, and as we investigate below the formal relations between Quarc and PC, we decided not to introduce it here.

We turn to the language of Quarc.[1]

2. Quarc: Syntax

The syntax of Quarc is a little simplified compared to that of [2], making less and somewhat different use of parentheses and commas. The formula rules are also somewhat different, in order to achieve unique parsing (as will be proved below), absent from the 2014 version. However, there is a straightforward bidirectional translation between the formulas of both versions (a fact we shall not prove here).

2.1. Language

Definition 2.1.1. (*Language*) A *language* of Quarc consists of:

- singular arguments: a non-empty, countable set of symbols, disjoint from the set of all other symbols listed below and strings thereof.
- for each $n > 0$, n-ary predicates: P_0^n, P_1^n, P_2^n, ...
- numerals: **1**, **2**, **3**, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$
- anaphors: x_0, x_1, x_2, ...
- quantifiers: $\forall$, $\exists$ (the *universal* and *particular* quantifiers)
- parentheses: (,)
- comma: ,

Note that anaphors will also be used as subscripts (see Definition 2.2.3), while numerals and the comma are used only as superscripts (see Definition 2.2.2).

Remark. We will use a, b, c, ... (possibly with subscripts) for arbitrary singular arguments, x, y, z, ... for arbitrary anaphors, and P, R, S, ... for arbitrary predicates. We will sometimes use Q for either of $\forall$ and $\exists$.

[1] For some of the formal work of this paper we consulted [7] and [34].

2.2. Formulas

DEFINITION 2.2.1. (*Quantified Arguments*) Let P be a unary predicate. Then $\forall P$ and $\exists P$ will be called (universally and particularly) *quantified arguments*.

DEFINITION 2.2.2. (*Reordered predicates*) Let P be an n-ary predicate ($n > 1$) and $\tau = \tau_1, \ldots, \tau_n$ a non-identity permutation of $\mathbf{1}, \ldots, \mathbf{n}$. Then P^τ is called a *reordered predicate* or *reordered form of* P.

For example, if R is a binary predicate, then $R^{\mathbf{2,1}}$ is a reordered (binary) predicate, and is *the* reordered form of R (as $\mathbf{2}, \mathbf{1}$ is the only non-identity permutation of $\mathbf{1}, \mathbf{2}$); if S is a ternary predicate, then $S^{\mathbf{2,3,1}}$ is one of the five reordered forms of S.

DEFINITION 2.2.3. (*Labels and Sources*) Let a be a singular argument, QP a quantified argument and x an anaphor. Then a_x and QP_x are called *x-labelled* (singular and quantified) arguments, where the x (written as a subscript) is considered not an anaphor but a label. In a string of symbols, *the source of an occurrence of an anaphor x* is the closest occurrence of an x-labelled argument to its left. If, in a string, α is the source of (an occurrence of) x, we say that (the occurrence of) x is an anaphor *of* α. A label of a singular or quantified argument is not part of the argument it is attached to.

DEFINITION 2.2.4. (*Formulas*) Let L be a Quarc language. The *formulas* of L (or L-formulas) are defined inductively as follows:

(a) Let $a_1, \ldots, a_n$ be unlabelled singular arguments and P a non-reordered n-ary predicate. Then $a_1 \ldots a_n P$ is a formula, which is also called a *basic* formula.

(b) Let $a_1 \ldots a_n P$ ($n > 1$) be a basic formula and $\tau = \tau_1, \ldots, \tau_n$ a non-identity permutation of $\mathbf{1}, \ldots, \mathbf{n}$. Then $a_{\tau_1} \ldots a_{\tau_n} P^\tau$ is a formula.

(c) Let $a_1 \ldots a_n P$ be a formula where none of $a_1, \ldots, a_n$ is labelled (but P may be reordered). Then $a_1 \ldots a_n \neg P$ is also a formula.

(d) Let ϕ be a formula. Then so is $\neg\phi$.

(e) Let ϕ and ψ be formulas. Then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$ and $(\phi \rightarrow \psi)$.

(f) Let ϕ be a formula containing distinct occurrences $o_1, \ldots, o_n$ ($n > 1$) of a singular argument a, where o_1 is the leftmost one among o_i and none of o_i is labelled. (ϕ may contain additional occurrences of a; and o_1 need not be the leftmost occurrence of a in ϕ.) Let x be an anaphor not occurring in ϕ. If to the left of o_1 there is no argument which is

a quantified argument or a labelled singular argument, and ϕ has no substring ψ which is a formula containing $o_1, \ldots, o_n$ and all anaphors of any argument occurring in it, then $\phi[a_x/o_1, x/o_2, \ldots, x/o_n]$ is a formula, and it is said to be *led* by a_x. ($\phi[a_x/o_1, x/o_2, \ldots, x/o_n]$ is got from ϕ by substituting a_x for o_1 and x for o_i $(1 < i \leqslant n)$.)

(g) Let ϕ be a formula containing an occurrence o of a singular argument, and QP an unlabelled quantified argument. If to the left of o there is no argument which is a quantified argument or a labelled singular argument, and ϕ contains no substring ψ which is a formula containing o and all anaphors of any argument occurring in it, then $\phi[QP/o]$ is a formula, and it is said to be *governed* by that occurrence of QP. ($\phi[QP/o]$ is got from ϕ by substituting QP for o.)

(h) Nothing else is a formula.

Remark. Formulas may contain only parentheses introduced in step (e), and we shall often omit parentheses where no ambiguity arises.

For example, $(aaR \wedge abR)$ is a Quarc formula where no anaphor or quantifier occurs, from which we can construct $(aa_xR \wedge xbR)$, which is a formula led by a_x. From the latter we can in turn construct $(a\forall P_xR \wedge xbR)$, which is a formula governed by (that occurrence of) $\forall P$. We may also write this formula without parentheses, $a\forall P_xR \wedge xbR$.

Proposition 2.2.5. (*Unique parsing*) *Let χ be a formula of Quarc. Then exactly one of the following is the case:*

(a) χ is a basic formula.

(b) χ is $a_1 \ldots a_n P^\tau$, where $a_1, \ldots, a_n$ are unlabelled singular arguments and P^τ is a reordered predicate.

(c) χ is $a_1 \ldots a_n \neg P$, where $a_1, \ldots, a_n$ are unlabelled and P may be reordered.

(d) χ has the form $\neg\phi$.

(e) χ has exactly one of the forms $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and it is neither led nor governed.

(f) χ is immediately generated from a formula ϕ by $[a_x/o_1, x/o_2, \ldots, x/o_n]$, as in 2.2.4(f), and is thus led by (that occurrence of) a_x.

(g) χ is immediately generated from a formula ϕ by $[QP/o]$, as in 2.2.4(g), and is thus governed by that occurrence of QP.

Moreover, in case (b), the basic formula from which χ is generated is uniquely determined; in case (c), $a_1 \ldots a_n P$ is a uniquely determined formula; in

case (d), ϕ is a uniquely determined formula; in case (e), both ϕ and ψ are uniquely determined formulas; in case (f), a_x and ϕ are uniquely determined; and in case (g), QP and ϕ (except for the singular argument at o) are uniquely determined.

Proof. It follows from 2.2.4 that at least one of (a) – (g) is the case. So what remains is the uniqueness claims. There are two sorts of uniqueness claim here. One is 'exactly one of (a) – (g) is the case' for χ, the other is about the uniqueness of the 'predecessor(s)' of χ in each of (a) – (g). We omit the proofs of such claims as they are straightforward. ■

With parsing being unique, we can proceed to define the complexity of formulas, which will later facilitate inductive proofs.

Definition 2.2.6. (*Complexity*) The *complexity*, *comp*, is a function from formulas of Quarc to $\mathbb{N}$. Let χ be a formula of Quarc. Then

(a) If χ is a basic formula, then $comp(\chi) = 0$.

(b) If χ is $a_1 \dots a_n P^\tau$, then $comp(\chi) = comp(a_1 \dots a_n P) + 1 (= 1)$.

(c) If χ is $a_1 \dots a_n \neg P$, then $comp(\chi) = comp(a_1 \dots a_n P) + 1$.

(d) If χ has the form $\neg\phi$, then $comp(\chi) = comp(\phi) + 1$.

(e) If χ has one of the forms $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and it is neither led nor governed, then $comp(\chi) = max\{comp(\phi), comp(\psi)\} + 1$.

(f) If χ is immediately generated from a formula ϕ in the way of 2.2.4(f), then $comp(\chi) = comp(\phi) + 1$.

(g) If χ is immediately generated from a formula ϕ in the way of 2.2.4(g), then $comp(\chi) = comp(\phi) + 1$.

3. Quarc: Truth-valuational Semantics

In [2], a truth-valuational semantics (TVS) was used, as was done in [3]. We shall use such semantics here too, and not the more familiar model-theoretic one. Truth-valuational semantics is intuitive, powerful and elegant, and of much philosophical interest. Significant formal work has been done on it, especially around the seventies [11,14,17–19], and it deserves more attention than it currently receives. It has met some philosophical criticisms, which we hope to address elsewhere (but see (Ben-Yami manuscript)), and a formal one [20], which was addressed in [3]. Although our purpose here is not to defend this approach, its successful application below may serve this aim

as well. — Our main results, however, can also be translated into model-theoretic semantics.

As was mentioned above, we shall explore both a two-valued version of Quarc and a three-valued one. The three-valued version follows Strawson's claim that, when the predicate in the quantified argument of a subject-predicate statement has no instances, it is natural to take that statement as lacking a truth value ([31] sec. V.c; [32], sec. 6.III.7). One who asserts, 'Some/all/most/seven students are present', or even 'No student is present', *presupposes*, on this analysis, that there are students. This presupposition supports the classification of statements of the form, All/some S are P, as lacking a truth value when S has no instances. And although the common approach in contemporary logic is to make the formal version of the universal 'Every S is P' vacuously true when S is as above, it is doubtful that this reflects truth in natural language: few would take the sentence,

7. All my children love their mother

to be true, when uttered by a childless person [15, 553]. Attempting to remain closer to this analysis of the semantics of natural language, both $\forall SP$ and $\exists SP$ will be defined below as neither true nor false in case S has no instances.

We shall generalise this gap approach to any formula of the form, $\alpha_1 \ldots \alpha_n$ P, in which any of the α_i is a quantified argument. This, we emphasise, is a *regimentation* of what is found in natural language (as Strawson himself later claimed [33]). Some sentences of this form may ordinarily be taken as false, while others as neither true nor false. However, some such regimentation seems unavoidable when applying exact formal tools to ordinary language; even if one doubts Strawson's remark, that 'ordinary language has no exact logic' [31, 344], it is enough that ordinary language has no logic capturable by an extensional system comparable in its simplicity to Quarc. All the same, the unavoidability of regimentation makes other alternatives also worth exploring, as was done for example in [24].

We shall make Quarc two-valued by imposing an instantiation rule, forcing each unary predicate to have instances, as was done in [2, 130]. The three-valued system will be obtained by eliminating this rule. We shall introduce a third value, *'undefined'* or u, to capture the cases discussed above and others dependent on them.

We proceed to the definitions.

3.1. Two-valued TVS

Definition 3.1.1. (*2-valuation*) For a Quarc language L, a *2-valuation* is a function, v, from the set of L-formulas to $\{0, 1\}$ that satisfies the following rules:

(a) **Basic formula.** For every basic formula ϕ, either $v(\phi) = 1$ or $v(\phi) = 0$.[2]

(b) **Reorder.** Let $a_1 \dots a_n P$ $(n > 1)$ be a basic formula and $\tau = \tau_1, \dots, \tau_n$ a non-identity permutation of $\mathbf{1}, \dots, \mathbf{n}$. Then: $v(a_{\tau_1} \dots a_{\tau_n} P^{\tau}) = v(a_1 \dots a_n P)$.

(c) **Negative predication.** Let $a_1 \dots a_n P$ be a formula where none of $a_1, \dots, a_n$ is labelled (but P may be reordered). Then: $v(a_1 \dots a_n \neg P) = 1$ if $v(a_1 \dots a_n P) = 0$; $v(a_1 \dots a_n \neg P) = 0$ if $v(a_1 \dots a_n P) = 1$.

(d) **Connectives.** Let ϕ and ψ be formulas. Then:

$v(\neg\phi) = 1$ if $v(\phi) = 0$; otherwise $v(\neg\phi) = 0$.
$v(\phi \wedge \psi) = 1$ if $v(\phi) = 1$ and $v(\psi) = 1$; otherwise $v(\phi \wedge \psi) = 0$.
$v(\phi \vee \psi) = 1$ if $v(\phi) = 1$ or $v(\psi) = 1$; otherwise $v(\phi \vee \psi) = 0$.
$v(\phi \rightarrow \psi) = 1$ if $v(\phi) = 0$ or $v(\psi) = 1$; otherwise $v(\phi \rightarrow \psi) = 0$.

(e) **Anaphora.** Let $\phi(a_x)$ be a formula led by a_x. Then $v(\phi(a_x)) = v(\phi(a))$, where $\phi(a)$ is the formula from which $\phi(a_x)$ is immediately generated.

(f) **Particular quantification.** Let $\phi(\exists P)$ be a formula governed by an occurrence of $\exists P$. Then: $v(\phi(\exists P)) = 1$ if $v(cP) = 1$ and $v(\phi(c)) = 1$ for some singular argument c in L; otherwise $v(\phi(\exists P)) = 0$. (When we write $\phi(QP)$ for a formula governed by an occurrence of a quantified argument QP, $\phi(c)$ is the result of replacing the governing occurrence of QP with a singular argument c.)

(g) **Universal quantification.** Let $\phi(\forall P)$ be a formula governed by an occurrence of $\forall P$. Then: $v(\phi(\forall P)) = 1$ if $v(\phi(c)) = 1$ for every c in L for which $v(cP) = 1$; otherwise $v(\phi(\forall P)) = 0$.

(h) **Instantiation.** For every *unary* predicate P in L, there is a singular argument a in L for which $v(aP) = 1$.

We sometimes write 'true' and 'false' instead of 1 and 0.

Remark. It follows from this definition and unique parsing (Proposition 2.2.5) that a valuation for L is uniquely determined by its assignment to

[2] Although this clause is formally redundant, we list it here to facilitate comparison with the three-valued system, which will include it as well.

basic formulas of L. Similarly for the three-valued semantics introduced later.

Validity on the truth-valuational approach is defined while allowing the addition and elimination of individual constants to and from a language; namely, validity is independent of a specific individual constant list [2, 131], [11, 183], [14,19]. Such a definition will be used below.

Definition 3.1.2. (*2-validity*) Let L be a Quarc language. An argument whose premises constitute the set Γ of L-formulas and whose conclusion is the L-formula ϕ is *2-valid*, written $\Gamma \vDash_2 \phi$, iff for any language which contains all the singular arguments occurring in either Γ or ϕ, every 2-valuation that assigns all formulas in Γ 'true' assigns ϕ 'true' as well.

Remark. If an argument in a Quarc language is valid, it is valid as an argument in every Quarc language which contains all the singular arguments involved. For this reason, we do not define validity as relative to specific languages.

3.2. Three-valued TVS

As mentioned above, the three-valued version is obtained by eliminating the Instantiation rule. As basic formulas remain either true or false and as the truth conditions specified in the valuation rules Reorder, Negative predication, and Anaphora are also not affected, we have to modify only the rules Connectives, where we shall use Kleene's strong tables, and Particular and Universal quantification:

Definition 3.2.1. (*3-valuation*) For a Quarc language L, a *3-valuation* is a function, v, from the set of L-formulas to $\{0, 1, u\}$ that satisfies the following rules (in addition to those same as in Definition 3.1.1):

(d) Connectives. Let ϕ and ψ be formulas. Then:

$v(\neg\phi) = 1$ if $v(\phi) = 0$; $v(\neg\phi) = 0$ if $v(\phi) = 1$; otherwise $v(\neg\phi) = u$.

$v(\phi \wedge \psi) = 1$ if $v(\phi) = 1$ and $v(\psi) = 1$; $v(\phi \wedge \psi) = 0$ if $v(\phi) = 0$ or $v(\psi) = 0$; otherwise $v(\phi \wedge \psi) = u$.

$v(\phi \vee \psi) = 1$ if $v(\phi) = 1$ or $v(\psi) = 1$; $v(\phi \vee \psi) = 0$ if $v(\phi) = 0$ and $v(\psi) = 0$; otherwise $v(\phi \vee \psi) = u$.

$v(\phi \rightarrow \psi) = 1$ if $v(\phi) = 0$ or $v(\psi) = 1$; $v(\phi \rightarrow \psi) = 0$ if $v(\phi) = 1$ and $v(\psi) = 0$; otherwise $v(\phi \rightarrow \psi) = u$.

(f) Particular quantification. Let $\phi(\exists P)$ be a formula governed by an occurrence of $\exists P$. Then:

$v(\phi(\exists P)) = 1$ if $v(cP) = 1$ and $v(\phi(c)) = 1$ for some singular argument c in L;

$v(\phi(\exists P)) = 0$ if $v(cP) = 1$ for some c in L but for every such c, $v(\phi(c)) = 0$;

$v(\phi(\exists P)) = u$ otherwise.

(g) Universal quantification. Let $\phi(\forall P)$ be a formula governed by an occurrence of $\forall P$. Then:

$v(\phi(\forall P)) = 1$ if $v(cP) = 1$ for some c in L and for every such c, $v(\phi(c)) = 1$;

$v(\phi(\forall P)) = 0$ if $v(cP) = 1$ and $v(\phi(c)) = 0$ for some c in L;

$v(\phi(\forall P)) = u$ otherwise.

Notice that a gap, u, can be *introduced* only by the two quantification rules. Rules (a), (b) and (c) never assign u to a formula and rules (d) and (e) assign u to a formula ϕ only if one is already assigned to the formula or formulas used to determine ϕ's truth value. Consequently, if every unary predicate does have an instance, no gap will be introduced at any stage and the rules will coincide with the 2-valuation rules. Hence,

Proposition 3.2.2. *Any 2-valuation is also a 3-valuation.*

While the conception of validity for a two-valued system is clear, namely, truth of the premises leads to truth of the conclusion, we face several formal options when we move to a three-valued system. We may keep the conception of truth leading to truth – *strict-to-strict* (SS) validity, as used in [15]; we may also consider an argument valid just in case, if its premises are not false, its conclusion isn't false either – *tolerant-to-tolerant* (TT) validity [12,27]; or we may require that truth does not lead to falsity – *strict-to-tolerant* (ST) validity; and other options are also possible.[3]

Each of these options has formal advantages and disadvantages. For instance, while SS validity is transitive, in the sense that if $\phi \vDash \psi$ and $\psi \vDash \chi$ then $\phi \vDash \chi$, the Deduction Theorem does not generally hold in it: if we use Kleene's strong tables, as above, then $p, q \vDash q$ but $p \nvDash q \rightarrow q$. (All the claims in this paragraph are easy to verify.) On TT validity with strong Kleene tables, modus ponens is invalid; while ST validity is not transitive. Formal considerations might weigh for or against a certain choice, but they cannot decide between them.

[3]The terms used here are taken from [8,9].

In this paper we adopt strict-to-tolerant (ST) validity. While Quarc with SS validity has already been researched [15], no publication has explored Quarc with ST validity. Moreover, as we shall see starting with the next subsection, Quarc with ST validity has several interesting formal properties. We therefore define:

Definition 3.2.3. (*3-validity*) Let L be a Quarc language. An argument whose premises constitute the set Γ of L-formulas and whose conclusion is the L-formula ϕ is *3-valid*, written $\Gamma \models_3 \phi$, iff for any language which contains all the singular arguments occurring in either Γ or ϕ, no 3-valuation assigns 'true' to all formulas in Γ and 'false' to ϕ.

From a *conceptual* point of view, ST validity allows one to represent possible failures of truth-preservation that do not lead to falsity as not being features of invalid argumentation. In a system in which false presupposition causes a truth value gap, as in the three-valued Quarc developed here, ST validity allows us to distinguish between invalid arguments and presupposition failure. This might reflect to some extent intuitive classifications: the argument,

8. All children love their mother; so, all Bob's children love their mother.

might seem 'alright', although on the approach developed above to presupposition, in case Bob has no children the conclusion isn't true even if the premise is. However, not all cases in which an argument is ST-valid despite presupposition failure are also intuitively 'alright'. For instance, the following two arguments, suggested by a reviewer,

9. Bob has no children; so, some of Bob's children love their mother.

10. Bob has no children; so, either some of Bob's children love their mother, or grass is green.

seem both objectionable. Accordingly, whether such a systematisation of the relations of validity to presupposition failure reflects anything intuitive, or whether it is just a formal way of distinguishing them, requires further consideration. But since the *formal* interest in ST-validity does not depend on the verdict on this question, we shall not pursue it any further here.

3.3. Coincidence of the Consequence Relations

In this subsection we prove Coincidence: any Quarc argument is 2-valid iff it is 3-valid. The basic idea behind this proof was already noted by [10],

comparing classical consequence (our $\models_2$) with strict-to-tolerant consequence (our $\models_3$):

> Obviously, a classical countermodel to the entailment from Γ to Δ is an st-countermodel. But conversely, any st-countermodel can be turned into a classical countermodel, basically because reassignments of the values 1 or 0 to subsentences with value $\frac{1}{2}$ in the original model do not alter the value 1 or 0 assigned to the sentences in which they appear. [10, 21]

However, that *there is* a classical model, or in our case a 2-valuation, which coincides with a given 3-valuation on the value it assigns to any formula which isn't assigned u on that 3-valuation, should be shown. We shall soon do that for Quarc, but we shall later consider a calculus in which this is *not* the case (§6). Namely, we shall later show that for some calculi, the classical consequence relation *does not* coincide with the strict-to-tolerant one. Here we proceed with the proof for Quarc.

PROPOSITION 3.3.1. *For any set* Γ *of L-formulas and L-formula* ϕ*, if* $\Gamma \models_3 \phi$ *then* $\Gamma \models_2 \phi$*.*

PROOF. We prove the contrapositive. Suppose $\Gamma \not\models_2 \phi$. Then, for some language L', there is a 2-valuation v on which all formulas in Γ are true and ϕ is false. Since v is also a 3-valuation (Proposition 3.2.2), $\Gamma \not\models_3 \phi$. ■

PROPOSITION 3.3.2. *For every Quarc language L, for every 3-valuation w for L, there is a language* L' *extending L (in the sense that every L-formula is an* L'*-formula) and there is a 2-valuation v for* L' *such that for every L-formula* ϕ*: (i) if* $w(\phi) = 1$ *then* $v(\phi) = 1$*; and (ii) if* $w(\phi) = 0$ *then* $v(\phi) = 0$*.*

PROOF. For each language L, let $L' = L \cup \{e\}$, where e is a singular argument new to L. For each 3-valuation w for L, let v be a 2-valuation for L' such that:

(a) for every basic L-formula ϕ, $v(\phi) = w(\phi)$;

(b) for every unary predicate P such that $w(aP) = 0$ for every a in L, $v(eP) = 1$;

(c) for every other basic L'-formula φ (i.e. those not decided in (a) or (b)), $v(\varphi) = 0$.

Given the definition above, every unary predicate has instances, and therefore v is indeed a 2-valuation. Then, the proposition is proved by induction on the complexity of L-formulas.

Base case: ϕ has complexity 0, in which case ϕ is a basic formula, then by definition $v(\phi) = w(\phi)$.

Induction step: ϕ has complexity $n + 1$, assuming that the proposition holds for all formulas of complexity up to n. Then the case is divided into subcases according to Proposition 2.2.5. Here we consider only two of them, the proofs of the rest adding nothing of interest.

(1) ϕ has the form $\neg\psi$. Suppose $w(\phi) = 1$. Then $w(\psi) = 0$; hence, by the induction hypothesis (*by IH* below), $v(\psi) = 0$; and hence $v(\phi) = 1$. Suppose $w(\phi) = 0$. Then, similarly, $v(\phi) = 0$.

(2) ϕ is governed by an occurrence of $\forall P$. Suppose $w(\phi) = 1$. Then $w(cP) = 1$ for some c in L and $w(\phi(c)) = 1$ for every such c; hence, by IH, $v(cP) = 1$ for some c in L' and $v(\phi(c)) = 1$ for every such c. In this case $v(eP) = 0$, hence $v(\phi(e))$ is irrelevant for $v(\phi)$, and hence $v(\phi) = 1$. Suppose $w(\phi) = 0$. Then, similarly, $v(\phi) = 0$.

■

PROPOSITION 3.3.3. *For any set* Γ *of L-formulas and L-formula* ϕ*, if* $\Gamma \vDash_2 \phi$ *then* $\Gamma \vDash_3 \phi$.

PROOF. We prove the contrapositive. Suppose $\Gamma \nvDash_3 \phi$. Then for some language L' there is a 3-valuation on which all formulas in Γ are true and ϕ is false; hence, by Proposition 3.3.2, for some language $L'' = L' \cup \{e\}$, there is a 2-valuation on which all formulas in Γ are true and ϕ is false. Hence, $\Gamma \nvDash_2 \phi$.

■

PROPOSITION 3.3.4. *(Coincidence) For any set* Γ *of formulas and formula* ϕ*,* $\Gamma \vDash_2 \phi$ *iff* $\Gamma \vDash_3 \phi$.

PROOF. By Propositions 3.3.1 and 3.3.3. ■

4. A Proof System

4.1. Lemmon-style Natural Deduction

DEFINITION 4.1.1. (*Proof*) A *proof* is a sequence of lines of the form $\langle L, (i), \phi \rangle$, where L is a possibly empty list of line numbers; (i) the *line number* in parenthesis; ϕ a formula of Quarc; and L and ϕ are written according to some *derivation rule* of those listed below in Definition 4.1.2, which is the *justification* for that line. ϕ is said to *depend* on the formulas in the lines listed in L. The line numbers in L are written without repetitions and in ascending

order. The formula in the last line of the proof is its *conclusion*. If there is a proof with the formula ϕ as conclusion, depending only on formulas from the set Γ, then ϕ is *provable* from Γ, written $\Gamma \vdash \phi$.

Remark. Since Quarc languages differ only in their singular arguments, any proof is a proof in any Quarc language that contains all the singular arguments occurring in it.

We shall usually write to the right of the formula the name of the derivation rule which justifies the line, possibly followed by line numbers, according to the conventions specified below.

Definition 4.1.2. (*Derivation rules*) Every Greek letter that occurs in this definition, if not otherwise specified, stands for a *formula*. We write L_1, L_2 for the list of all numbers occurring in L_1 or L_2, and $L - i$ for the result of removing i from L (if it occurs in it).

Premise

i	(i)	ϕ	Premise

Negation Introduction (¬I)

i	(i)	ϕ	Premise
L_1	(j)	ψ	
L_2	(k)	$\neg\psi$	
	$\vdots$		
$(L_1, L_2) - i$	(l)	$\neg\phi$	$\neg$I i, j, k

Negation Elimination (¬E)

L	(i)	$\neg\neg\phi$	
	$\vdots$		
L	(j)	ϕ	$\neg$E i

Conjunction Introduction (∧I)

L_1	(i)	ϕ	
L_2	(j)	ψ	
	$\vdots$		
L_1, L_2	(k)	$(\phi \wedge \psi)$	$\wedge$I i, j

Conjunction Elimination (∧E)

$$\begin{array}{llll} L & (i) & (\phi \wedge \psi) & \\ & \vdots & & \\ L & (j) & \phi & \wedge\text{E } i \end{array} \qquad \begin{array}{llll} L & (i) & (\phi \wedge \psi) & \\ & \vdots & & \\ L & (j) & \psi & \wedge\text{E } i \end{array}$$

Disjunction Introduction (∨I)

$$\begin{array}{llll} L & (i) & \phi & \\ & \vdots & & \\ L & (j) & (\phi \vee \psi) & \vee\text{I } i \end{array} \qquad \begin{array}{llll} L & (i) & \psi & \\ & \vdots & & \\ L & (j) & (\phi \vee \psi) & \vee\text{I } i \end{array}$$

Disjunction Elimination (∨E)

$$\begin{array}{llll} L_1 & (i) & (\phi \vee \psi) & \\ j & (j) & \phi & \text{Premise} \\ L_2 & (k) & \chi & \\ l & (l) & \psi & \text{Premise} \\ L_3 & (m) & \chi & \\ & \vdots & & \\ L_1, (L_2 - j), (L_3 - l) & (n) & \chi & \vee\text{E } i, j, k, l, m \end{array}$$

Conditional Introduction (→I)

$$\begin{array}{llll} i & (i) & \phi & \text{Premise} \\ L & (j) & \psi & \\ & \vdots & & \\ L - i & (k) & (\phi \rightarrow \psi) & \rightarrow\text{I } i, j \end{array}$$

Conditional Elimination (→E)

$$\begin{array}{llll} L_1 & (i) & (\phi \rightarrow \psi) & \\ L_2 & (j) & \phi & \\ & \vdots & & \\ L_1, L_2 & (k) & \psi & \rightarrow\text{E } i, j \end{array}$$

Reorder Introduction (RI)

$$\begin{array}{llll} L & (i) & a_1 \dots a_n P & \\ & \vdots & & \\ L & (j) & a_{\tau_1} \dots a_{\tau_n} P^{\tau} & \text{RI } i \end{array}$$

In this and the next rule, $a_1 \dots a_n P$ $(n > 1)$ is a basic formula and $\tau = \tau_1, \dots, \tau_n$ is a non-identity permutation of $\mathbf{1}, \dots, \mathbf{n}$.

Reorder Elimination (RE)

$$\begin{array}{llll} L & (i) & a_{\tau_1} \dots a_{\tau_n} P^{\tau} & \\ & \vdots & & \\ L & (j) & a_1 \dots a_n P & \text{RE } i \end{array}$$

Sentence negation to predicate negation (SP)

$$\begin{array}{llll} L & (i) & \neg a_1 \dots a_n P & \\ & \vdots & & \\ L & (j) & a_1 \dots a_n \neg P & \text{SP } i \end{array}$$

In this and the next rule, P may be reordered.

Predicate negation to sentence negation (PS)

$$\begin{array}{llll} L & (i) & a_1 \dots a_n \neg P & \\ & \vdots & & \\ L & (j) & \neg a_1 \dots a_n P & \text{PS } i \end{array}$$

Anaphor Introduction (AI)

$$\begin{array}{llll} L & (i) & \phi(a) & \\ & \vdots & & \\ L & (j) & \phi(a_x) & \text{AI } i \end{array}$$

In this and the next rule, $\phi(a_x)$ is a formula led by a_x; $\phi(a)$ is the formula from which $\phi(a_x)$ is immediately generated.

Anaphor Elimination (AE)

$$\begin{array}{llll} L & (i) & \phi(a_x) & \\ & \vdots & & \\ L & (j) & \phi(a) & \text{AE } i \end{array}$$

Universal Elimination (∀E)

L_1	(i)	$\phi(\forall P)$	
L_2	(j)	aP	
		$\vdots$	
L_1, L_2	(k)	$\phi(a)$	$\forall$E i, j

In this and the next three rules, $\phi(QP)$ is a formula governed by an occurrence of QP, and $\phi(a)$ is the formula got from $\phi(QP)$ by replacing *that* occurrence of QP with a.

Universal Introduction (∀I)

i	(i)	aP	Premise
L	(j)	$\phi(a)$	
		$\vdots$	
$L - i$	(k)	$\phi(\forall P)$	$\forall$I i, j

In this rule, a must not occur in $\phi(\forall P)$ or any formula in lines $L - i$.

Particular Introduction (∃I)

L_1	(i)	aP	
L_2	(j)	$\phi(a)$	
		$\vdots$	
L_1, L_2	(k)	$\phi(\exists P)$	$\exists$I i, j

Instantial Import (Imp)

L_1	(i)	$\phi(QP)$	
j	(j)	aP	Premise
k	(k)	$\phi(a)$	Premise
L_2	(l)	ψ	
		$\vdots$	
$L_1, (L_2 - j - k)$	(m)	ψ	Imp i, j, k, l

In this rule, a must not occur in $\phi(QP)$, ψ, any formula in lines L_1, or any formula in lines $L_2 - j - k$.

4.2. Some Useful Examples

The following two examples will be used in the completeness proof in section 5.

Example 4.2.1. The following proof shows that $\vdash \exists PP$.

1	(1)	aP	Premise
	(2)	$\forall PP$	∀I 1,1
1	(3)	$\exists PP$	∃I 1,1
	(4)	$\exists PP$	Imp 2,1,1,3

We next show that $\neg\phi(\forall P) \vdash \exists P_x P \wedge \neg\phi(x)$ (a more exact formulation is provided below). Together with the proof of the other direction, which we shall not provide here, and given the soundness of the system (see below), this shows an equivalence both in the proof system and the semantics between universal and particular quantification in Quarc, namely the equivalence of the formulas $\neg\phi(\forall P)$ and $\exists P_x P \wedge \neg\phi(x)$.

Example 4.2.2. Let $\phi(\forall P)$ be a formula governed by an occurrence of $\forall P$. Assuming that occurrence of $\forall P$ is not labelled, let x be an anaphor that does not occur in $\phi(\forall P)$, and $\phi(x)$ the string got from $\phi(\forall P)$ by replacing that occurrence of $\forall P$ with x; let c be a singular argument that does not occur in $\phi(\forall P)$, and $\phi(c)$ the formula got from $\phi(\forall P)$ by replacing that occurrence of $\forall P$ with c. The following proof shows that $\neg\phi(\forall P) \vdash \exists P_x P \wedge \neg\phi(x)$. [4]

1	(1)	$\neg\phi(\forall P)$	Premise
2	(2)	$\neg(\exists P_x P \wedge \neg\phi(x))$	Premise
3	(3)	cP	Premise
4	(4)	$\neg\phi(c)$	Premise
3,4	(5)	$cP \wedge \neg\phi(c)$	∧I 3,4
3,4	(6)	$c_x P \wedge \neg\phi(x)$	AI 5
3,4	(7)	$\exists P_x P \wedge \neg\phi(x)$	∃I 3,6
2,3	(8)	$\neg\neg\phi(c)$	¬I 4,7,2
2,3	(9)	$\phi(c)$	¬E 8
2	(10)	$\phi(\forall P)$	∀I 3,9
1	(11)	$\neg\neg(\exists P_x P \wedge \neg\phi(x))$	¬I 2,10,1
1	(12)	$\exists P_x P \wedge \neg\phi(x)$	¬E 11

5. Completeness of Quarc

The soundness of the closely related proof system of [2] was proved in that paper, and the adaptation of that proof to the two-valued system of this

[4] The same proof is found in [22, p. 16] In case the governing occurrence of $\forall P$ in $\phi(\forall P)$ is labelled, the definitions of $\phi(x)$ and $\phi(c)$ are slightly different, as is the proof. That case won't be considered here.

paper is straightforward, so we do not provide it here. And since, by Coincidence, if $\Gamma \vDash_2 \phi$ then $\Gamma \vDash_3 \phi$, the proof system is also sound with respect to the 3-valued semantics.

An indirect proof of the completeness of Quarc with natural deduction is found in [24,25], taken together: a Gentzen-style proof theory is developed in these papers, in the former the authors show it to be equivalent to the proof system of [2], and in the latter they prove its completeness. However, a direct proof of the above hasn't been published in any article. A Henkin-style proof is found in [22] and in [6].[5]

In this section we provide a direct proof of the completeness of Quarc with natural deduction. First, we prove the completeness theorem for the two-valued Quarc: for any set Γ of formulas and formula ϕ, if $\Gamma \vDash_2 \phi$ then $\Gamma \vdash \phi$. The proof is an adaptation of Lindenbaum's construction to Quarc with truth-valuational semantics, and it is close to Leblanc's proof of the completeness of the first-order Predicate Calculus, where he also uses truth-valuational semantics [18, §2.3]. Then, by Coincidence, we will have the completeness of three-valued Quarc: if $\Gamma \vDash_3 \phi$ then $\Gamma \vdash \phi$. All the semantic concepts mentioned in this section, if not otherwise specified, are those of the two-valued Quarc.

5.1. Satisfiability and Consistency

We start with a couple of definitions:

Definition 5.1.1. (*Satisfiability*) A set Γ of formulas is *satisfiable* iff, for some language which contains all the singular arguments occurring in Γ, there is a valuation on which all formulas in Γ are true; we say of such a valuation that it *satisfies* Γ.

Definition 5.1.2. (*Consistency*) A set Γ of formulas is *consistent* iff, for any formula ϕ, at most one of ϕ and $\neg\phi$ is provable from Γ. A set of formulas is *inconsistent* if it is not consistent.

In the rest of this section, we provide the proof of the proposition below, from which the completeness theorem follows.

Proposition 5.1.3. *If a set of formulas is consistent then it is satisfiable.*

[5] Theorem 4 (p. 8) in the former work, which is essential for the completeness proof, had a small lacuna, fixed in the latter work (p. 63).

5.2. Maximal Consistent Set

DEFINITION 5.2.1. (*Maximal consistent set*) Δ is a *maximal consistent set* of L-formulas iff Δ is consistent and for every L-formula $\psi \notin \Delta$, $\Delta \cup \{\psi\}$ is inconsistent.

The following five propositions are consequences of this definition, whose proofs are straightforward and not provided here.

PROPOSITION 5.2.2. *Let Δ be a maximal consistent set of L-formulas. Then for every L-formula ϕ, if $\Delta \vdash \phi$ then $\phi \in \Delta$.*

PROPOSITION 5.2.3. *Let Δ be a maximal consistent set of L-formulas. Then for any L-formulas ϕ and ψ:*

(a) $\phi \in \Delta$ iff $\neg\phi \notin \Delta$.

(b) $\phi \wedge \psi \in \Delta$ iff $\phi \in \Delta$ and $\psi \in \Delta$.

(c) $\phi \vee \psi \in \Delta$ iff $\phi \in \Delta$ or $\psi \in \Delta$.

(d) $\phi \rightarrow \psi \in \Delta$ iff $\phi \notin \Delta$ or $\psi \in \Delta$.

PROPOSITION 5.2.4. *Let Δ be a maximal consistent set of L-formulas. Let $a_1, \ldots, a_n$ $(n > 1)$ be singular arguments in L, P a non-reordered n-ary predicate in L, and $\tau = \tau_1, \ldots, \tau_n$ a non-identity permutation of $\mathbf{1}, \ldots, \mathbf{n}$. Then:*

$$a_{\tau_1} \ldots a_{\tau_n} P^{\tau} \in \Delta \text{ iff } a_1 \ldots a_n P \in \Delta.$$

PROPOSITION 5.2.5. *Let Δ be a maximal consistent set of L-formulas. Let $a_1, \ldots, a_n$ be singular arguments in L, and P an n-ary predicate in L. Then:*

$$a_1 \ldots a_n \neg P \in \Delta \text{ iff } a_1 \ldots a_n P \notin \Delta.$$

PROPOSITION 5.2.6. *Let Δ be a maximal consistent set of L-formulas, and $\phi(a_x)$ an L-formula led by a_x. Then:*

$\phi(a_x) \in \Delta$ iff $\phi(a) \in \Delta$, where $\phi(a)$ is the formula from which $\phi(a_x)$ is immediately generated.

5.3. Instance and Witness

DEFINITION 5.3.1. Δ is an *instance-complete* set of L-formulas iff for each *unary* predicate P in L, $aP \in \Delta$ for some singular argument a in L.

DEFINITION 5.3.2. Δ is a *witness-complete* set of L-formulas iff for every L-formula $\phi(\exists P) \in \Delta$, where $\phi(\exists P)$ is governed by an occurrence of $\exists P$, there is a singular argument c such that $cP \in \Delta$ and $\phi(c) \in \Delta$.

PROPOSITION 5.3.3. *Let Δ be a maximal consistent and witness-complete set of L-formulas, and let $\phi(\exists P)$ and $\psi(\forall P)$ be L-formulas governed by occurrences of $\exists P$ and $\forall P$ respectively. Then:*

(a) $\phi(\exists P) \in \Delta$ iff for some c, $cP \in \Delta$ and $\phi(c) \in \Delta$.

(b) $\psi(\forall P) \in \Delta$ iff for every c for which $cP \in \Delta$, $\psi(c) \in \Delta$.

PROOF. (a) Suppose $\phi(\exists P) \in \Delta$. Then, by Definition 5.3.2, $cP \in \Delta$ and $\phi(c) \in \Delta$ for some c. Suppose $cP \in \Delta$ and $\phi(c) \in \Delta$ for some c. Then $\Delta \vdash cP$ and $\Delta \vdash \phi(c)$, hence by $\exists$I $\Delta \vdash \phi(\exists P)$, and hence $\phi(\exists P) \in \Delta$.

(b) Suppose $\psi(\forall P) \in \Delta$. Then by $\forall$E, for any singular argument c in L, if $cP \in \Delta$ then $\psi(c) \in \Delta$. For the other direction we prove the contrapositive. Suppose $\psi(\forall P) \notin \Delta$. Then $\neg\psi(\forall P) \in \Delta$, hence by Example 4.2.2 (assuming that the governing occurrence of $\forall P$ is not labelled) $\exists P_x P \wedge \neg\psi(x) \in \Delta$, hence by (a) $cP \in \Delta$ and $c_x P \wedge \neg\psi(x) \in \Delta$ for some c, hence $cP \in \Delta$ and $\neg\psi(c) \in \Delta$ for some c, and hence $cP \in \Delta$ and $\psi(c) \notin \Delta$ for some c. (The case in which the occurrence of $\forall P$ is labelled requires a simple adaptation of Example 4.2.2.) ■

5.4. Lindenbaum's Lemma

PROPOSITION 5.4.1. *(Lindenbaum's Lemma) Every consistent set Δ of L-formulas can be extended to a maximal consistent, instance- and witness-complete set of L^*-formulas, where L^* is obtained by adding denumerably many singular arguments to L.*

PROOF. Let $L^* = L \cup \{d_k : k \in \mathbb{N}\}$, where no d_k is in L and $d_i \neq d_j$ whenever $i \neq j$; and let $\phi_0, \phi_1, \phi_2, \ldots$ be an enumeration of all the formulas of L^*.

Suppose Δ is a consistent set of L-formulas. We construct a sequence $\Delta_0, \Delta_1, \Delta_2, \ldots$ of sets of L^*-formulas in the following scheme.

Let $\Delta_0 = \Delta$; and for each $n \in \mathbb{N}$:

(1) in case $\Delta_n \cup \{\phi_n\}$ is inconsistent, let $\Delta_{n+1} = \Delta_n$;

(2) in case $\Delta_n \cup \{\phi_n\}$ is consistent and ϕ_n is not governed by any occurrence of a particularly quantified argument, let $\Delta_{n+1} = \Delta_n \cup \{\phi_n\}$;

(3) in case $\Delta_n \cup \{\phi_n\}$ is consistent and ϕ_n is governed by an occurrence of a particularly quantified argument $\exists P$, let $\Delta_{n+1} = \Delta_n \cup \{\phi_n\} \cup \{dP\} \cup \{\phi(d)\}$, where d is the first item in the sequence $d_0, d_1, \ldots$ that does not occur in Δ_n or ϕ_n. $\phi(d)$ is the formula which results from ϕ_n by replacing that occurrence of $\exists P$ with d.

PROPOSITION 5.4.2. *Each set Δ_i in the sequence is consistent.*

PROOF. By induction on i in Δ_i.

Base case: Since $\Delta_0 = \Delta$ and Δ is consistent, Δ_0 is consistent.

Induction step: Assuming that Δ_n is consistent, we show that Δ_{n+1} is also consistent whichever case it falls into:

(1) $\Delta_{n+1} = \Delta_n$. Since Δ_n is consistent, so is Δ_{n+1}.

(2) $\Delta_{n+1} = \Delta_n \cup \{\phi_n\}$. In this case $\Delta_n \cup \{\phi_n\}$ is consistent, so Δ_{n+1} is consistent.

(3) $\Delta_{n+1} = \Delta_n \cup \{\phi_n\} \cup \{dP\} \cup \{\phi(d)\}$, in which case $\Delta_n \cup \{\phi_n\}$ is consistent and ϕ_n is governed by an occurrence of $\exists P$. We write $\phi(\exists P)$ for ϕ_n. Suppose for reductio that Δ_{n+1} is inconsistent. Then: $\Delta_n, \phi(\exists P), dP, \phi(d) \vdash \neg\phi(\exists P)$. Since d does not occur in Δ_n and $\phi(\exists P)$ (and hence does not occur in $\neg\phi(\exists P)$ either), by the rule Instantial Import we have $\Delta_n, \phi(\exists P) \vdash \neg\phi(\exists P)$. Hence, $\Delta_n \cup \{\phi_n\}$ is inconsistent, and we have a contradiction. ■

Now, let $\Delta^* = \bigcup_{i \in \mathbb{N}} \Delta_i$. It is easy to see that

PROPOSITION 5.4.3. *Δ^* is a maximal consistent set of L^*-formulas.*

Also, the construction ensures Δ^* is instance- and witness-complete, as proved respectively below.

PROPOSITION 5.4.4. *Δ^* is an instance-complete set of L^*-formulas.*

PROOF. For every unary predicate P in L^*, the formula $\exists PP$ is ϕ_n for some n. We already saw in Example 4.2.1 that $\vdash \exists PP$ for any unary predicate P, so $\Delta_n \cup \{\exists PP\}$ is consistent if Δ_n is consistent; and since $\exists PP$ is governed by that occurrence of $\exists P$, $\Delta_{n+1} = \Delta_n \cup \{\exists PP\} \cup \{dP\}$ for some d. Hence, for every unary predicate P in L^*, there is some d for which $dP \in \Delta^*$. ■

PROPOSITION 5.4.5. *Δ^* is a witness-complete set of L^*-formulas.*

PROOF. Suppose $\phi(\exists P)$ is a formula governed by $\exists P$ and $\phi(\exists P) \in \Delta^*$. Since Δ^* is consistent, $\Delta_n \cup \{\phi(\exists P)\}$ is consistent for any n. Let $\phi(\exists P)$ be ϕ_m. Then $\Delta_m \cup \phi(\exists P)$ is consistent, hence $dP \in \Delta_{m+1}$ and $\phi(d) \in \Delta_{m+1}$ for some d, and hence $dP \in \Delta^*$ and $\phi(d) \in \Delta^*$ for some d. ■

Now that Δ^* is maximal consistent, instance- and witness-complete, we have proved Lindenbaum's Lemma. ■

5.5. Truth Lemma

PROPOSITION 5.5.1. *Let L be a Quarc language and B an instance-complete set of basic L-formulas. If a 3-valuation v is such that, for any basic L-formula, $v(\phi) = 1$ iff $\phi \in B$, then v is a 2-valuation.*

PROOF. Since B is instance-complete, v complies with the Instantiation rule and is thus a 2-valuation. ■

PROPOSITION 5.5.2. *(Truth lemma) Let Δ be a maximal consistent, instance- and witness-complete set of L-formulas. Let v^* be the valuation for L such that: for every* basic *formula ϕ of L, $v^*(\phi) = 1$ iff $\phi \in \Delta$. Then, for every L-formula ϕ, $v^*(\phi) = 1$ iff $\phi \in \Delta$. (Since Δ is instance-complete, by 5.5.1 v^* is a 2-valuation.)*

PROOF. By induction on the complexity of L-formulas.

Base case: ϕ has complexity 0, in which case it is a basic formula, then by definition $v^*(\phi) = 1$ iff $\phi \in \Delta$.

Induction step: ϕ has complexity $n + 1$, where we assume (IH) that the proposition holds for every formula of complexity up to n. We consider the following subcases, the proofs of the rest adding nothing of interest.

(1) ϕ is $a_{\tau_1} \dots a_{\tau_n} P^{\tau}$. Then:

$v^*(\phi) = 1$ iff $v^*(a_1 \dots a_n P) = 1$
iff $a_1 \dots a_n P \in \Delta$ (IH)
iff $\phi \in \Delta$ (5.2.4)

(2) ϕ is $a_1 \dots a_n \neg P$. Then:

$v^*(\phi) = 1$ iff $v^*(a_1 \dots a_1 P) = 0$
iff $a_1 \dots a_n P \notin \Delta$ (IH)
iff $\phi \in \Delta$ (5.2.5)

(3) ϕ has the form $\neg\psi$. Then:

$v^*(\phi) = 1$ iff $v^*(\psi) = 0$
iff $\psi \notin \Delta$ (IH)
iff $\phi \in \Delta$ (5.2.3(a))

(4) ϕ has the form $\psi \wedge \chi$ and it is neither led nor governed. Then:

$v^*(\phi) = 1$ iff $v^*(\psi) = 1$ and $v^*(\chi) = 1$
iff $\psi \in \Delta$ and $\chi \in \Delta$ (IH)
iff $\phi \in \Delta$ (5.2.3(b))

(5) ϕ is led by a_x. We write $\phi(a)$ for the formula from which ϕ is immediately generated, so $\phi(a)$ is of complexity n. Then:

$$\begin{array}{lll} v^*(\phi) = 1 & \text{iff} & v^*(\phi(a)) = 1 \\ & \text{iff} & \phi(a) \in \Delta \qquad \text{(IH)} \\ & \text{iff} & \phi \in \Delta \qquad \text{(5.2.6)} \end{array}$$

(6) ϕ is governed by an occurrence of $\exists P$. We write $\phi(c)$ for the formula got from ϕ by replacing that occurrence with c, so $\phi(c)$ has complexity n. Then:

$$\begin{array}{lll} v^*(\phi) = 1 & \text{iff} & \text{for some } c,\ v^*(cP) = 1 \text{ and } v^*(\phi(c)) = 1 \\ & \text{iff} & \text{for some } c,\ cP \in \Delta \text{ and } \phi(c) \in \Delta \qquad \text{(IH)} \\ & \text{iff} & \phi \in \Delta \qquad \text{(5.3.3(a))} \end{array}$$

Thus, we have proved that for every L-formula ϕ, $v^*(\phi) = 1$ iff $\phi \in \Delta$. This shows that every maximal consistent, instance- and witness-complete set of formulas is satisfiable. ■

5.6. Summary

As we noted earlier, the completeness theorem follows from the proposition that every consistent set is satisfiable, i.e. for any consistent set Δ of L-formulas, there exists a valuation on which all formulas in Δ are true. (The valuation in question can be a valuation for any Quarc language which contains all the singular arguments occurring in Δ). We have shown by Lindenbaum's construction that every consistent set Δ of L-formulas can be extended to a maximal consistent, instance- and witness-complete set Δ^* of L^*-formulas (where L^* is L plus a list of new singular arguments). And, by the Truth Lemma, every such set Δ^* is satisfied by a valuation for L^*. Since Δ is a subset of Δ^*, the valuation also satisfies Δ. Thus, we have proved the completeness of two-valued Quarc:

Proposition 5.6.1. *(Completeness) For any set* Γ *of formulas and formula* ϕ*, if* $\Gamma \vDash_2 \phi$ *then* $\Gamma \vdash \phi$.

By the coincidence theorem we also have the completeness of three-valued Quarc:

Proposition 5.6.2. *(Completeness) For any set* Γ *of formulas and formula* ϕ*, if* $\Gamma \vDash_3 \phi$ *then* $\Gamma \vdash \phi$.

6. PC-to-Quarc Translation

The relations of Quarc and PC have been a topic of research since the first publication of a Quarc's precursor in [16], and several additional results have been established since [15,25,29]. We are readdressing the question here

because of the specific properties of the Quarc system of this paper. First, it has a three-valued truth-valuational semantics, unlike any system considered so far: all were two-valued, apart from the one of [15], which however used model-theoretic semantics. Moreover, unlike the three-valued system of [15], the system of this paper uses ST validity and, again unlike that system, it is not extended with defining clauses, which played an essential role in the PC-to-Quarc translation of that paper.

This section is dedicated to a general method for translating formulas of PC into formulas of Quarc. It also investigates whether such a translation preserves truth values and (in)validity.

6.1. PC: Syntax

In this subsection we introduce a version of the syntax of the Predicate Calculus.

Definition 6.1.1. (*Language*) A *language* of PC consists of:

- a non-empty countable set of constants.
- for every $n > 0$, n-ary predicates: P_0^n, P_1^n, P_2^n, ...
- variables: x_0, x_1, x_2, ...
- connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$.
- quantifiers: $\forall$, $\exists$.
- parenthesis: (,).

Remark. We will use a, b, c, ... (possibly with subscripts) for arbitrary constants, x, y, z, ... for arbitrary variables, and P, R, S, ... for arbitrary predicates. We will sometimes use Q for either of $\forall$ and $\exists$.

Definition 6.1.2. (*Formulas*) The *formulas* of PC are defined inductively as follows:

(a) If P is an n-ary predicate and $a_1, \ldots, a_n$ are constants, then $Pa_1 \ldots a_n$ is a formula, which is also called a *basic* formula. (Basic formulas do not contain variables.)

(b) If ϕ and ψ are formulas, then $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$ and $(\phi \rightarrow \psi)$ are also formulas.

(c) If ϕ is a formula containing one or more occurrences of a, and x is new to ϕ, then $\exists x\phi[x/a]$ and $\forall x\phi[x/a]$ are formulas. ($\phi[x/a]$ is the result of replacing every occurrence of a in ϕ with x.)

(d) Nothing else is a formula.

Complexity of PC formulas can be defined along the same lines as that of Quarc formulas (Definition 2.2.6).

6.2. Two-valued TVS for PC

Since we work with TVS for Quarc, it is simpler to work with TVS for PC as well. The set of truth-valuationally valid PC arguments can be shown to coincide with that of model-theoretically valid ones, along the lines of proofs found in [18, §4.2]. Accordingly, the preservation of validity under translation proved below applies also to PC with model-theoretic semantics.

Definition 6.2.1. (*2-valuation*) For a PC language L, a *2-valuation* is a function v from L-formulas to $\{0, 1\}$ such that:

(a) For each basic formula ϕ, either $v(\phi) = 1$ or $v(\phi) = 0$.

(b) Let ϕ and ψ be formulas of L. Then:

$v(\neg\phi) = 1$ if $v(\phi) = 0$; otherwise $v(\neg\phi) = 0$.
$v(\phi \wedge \psi) = 1$ if $v(\phi) = 1$ and $v(\psi) = 1$; otherwise $v(\phi \wedge \psi) = 0$.
$v(\phi \vee \psi) = 1$ if $v(\phi) = 1$ or $v(\psi) = 1$; otherwise $v(\phi \vee \psi) = 0$.
$v(\phi \rightarrow \psi) = 1$ if $v(\phi) = 0$ or $v(\psi) = 1$; otherwise $v(\phi \rightarrow \psi) = 0$.

(c) Let ϕ be a formula containing c and x a variable new to ϕ. Then:

$v(\exists x\phi[x/c]) = 1$ if $v(\phi[d/c]) = 1$ for some d in L; otherwise $v(\exists x\phi[x/c]) = 0$.
$v(\forall x\phi[x/c]) = 1$ if $v(\phi[d/c]) = 1$ for every d in L; otherwise $v(\forall x\phi[x/c]) = 0$.

Definition 6.2.2. (*2-validity*) Let L be a PC language. An argument whose premises constitute the set Γ of L-formulas and whose conclusion is the L-formula ϕ is *2-valid*, written $\Gamma \vDash_2 \phi$, iff for any PC language which contains all the constants occurring in either Γ or ϕ, every 2-valuation that assigns all formulas in Γ 'true' assigns ϕ 'true' as well.

6.3. The Translation Manual

As in some previous versions, the PC-to-Quarc translation involves the introduction into Quarc of a special unary predicate, T (for *Thing*), which is used to reflect in Quarc the fact that quantification in PC is not restricted by means of any predicate. Here we use T as a *logical* predicate, so it occurs as a constant in both semantics and proof system.[6]

[6]The specificities of T's incorporation differ between different works. In [15] it is not a logical predicate; in [25], it is first introduced only as part of a quantified phrase, $\forall T$ or $\exists T$.

We add a valuation rule to Definitions 3.1.1 (2-valuation) and 3.2.1 (3-valuation), which specifies the semantic behaviour of T:

(i) Thing. For any singular argument a, $v(aT) = 1$.

We proceed to correlate the PC and Quarc languages, where all Quarc languages we consider are enriched by the logical unary predicate T.

DEFINITION 6.3.1. (*Language correlation*) With each PC language L_p we correlate the unique Quarc language L_q that satisfies the following requirements:

(a) The singular arguments of L_q are the constants of L_p.

(b) The non-logical predicates of L_q are those of L_p, and arities are preserved.

Clearly, in this way every Quarc language is correlated with a unique PC language.

We define the PC-to-Quarc translation as a function that maps each and every formula of L_p to a formula of L_q.

DEFINITION 6.3.2. (*Translation function*) The translation function f is defined recursively as follows:

(1) *Basic formulas:*

$f(Pa_1 \dots a_n) = a_1 \dots a_n P$

(2) *Truth functional compounds:*

$f(\neg\phi) = \neg f(\phi)$
$f((\phi * \psi)) = (f(\phi) * f(\psi))$, where $*$ stands for one of $\wedge$, $\vee$ and $\rightarrow$.

(3) *Quantified formulas:*

$f(\forall x\phi[x/a]) = (\forall T_x T \wedge f(\phi)[x/a])$
$f(\exists x\phi[x/a]) = (\exists T_x T \wedge f(\phi)[x/a])$

The following proposition is easy to prove and will be used in later proofs.

PROPOSITION 6.3.3. *Let ϕ be a PC formula and $f(\phi)[d/c]$ the Quarc formula which results from replacing every occurrence of c in $f(\phi)$ with d. Then:* $f(\phi)[d/c] = f(\phi[d/c])$.

6.4. Truth Value Preservation

We move on to show that the PC-to-Quarc translation is adequate, in the sense that there is a bijection between PC valuations and Quarc 3-valuations

(where again, all Quarc languages we consider are enriched with T), such that the truth value of a formula is preserved by the translation under this bijection. It will follow that the (in)validity of PC-arguments is also preserved by the translation.

Definition 6.4.1. (*Valuation correlation*) With each PC valuation v_p for a PC language L_p we correlate a Quarc valuation v_q for L_q, the Quarc language that we correlate with L_p, such that for every *basic* formula ϕ of L_p, $v_q(f(\phi)) = v_p(\phi)$.

The following proof shows that the PC-to-Quarc translation, together with the valuation correlation, preserves truth values.

Proposition 6.4.2. *Let χ be a formula of a PC language L_p and $f(\chi)$ its translation in L_q, the Quarc language that we correlate with L_p. Let v_p be a valuation for L_p and v_q the correlated valuation for L_q. Then: $v_q(f(\chi)) = 1$ iff $v_p(\chi) = 1$; $v_q(f(\chi)) = 0$ iff $v_p(\chi) = 0$.*

Remark. We need to check both 'true' and 'false' cases, because Quarc formulas can also have the third value.

Proof. By induction on formulas of L_p.

Base case: Suppose χ is a basic formula of L_p. Then by the valuation correlation $v_q(f(\chi)) = 1$ iff $v_p(\chi) = 1$; and $v_q(f(\chi)) = 0$ iff $v_p(\chi) = 0$.

Induction step: Assuming that the proposition holds for all formulas less complex than χ, we consider the following cases:

(1) χ is $\neg\phi$, so $f(\chi) = \neg f(\phi)$. Suppose $v_p(\chi) = 1$. Then $v_p(\phi) = 0$; hence, by IH, $v_q(f(\phi)) = 0$; and hence $v_q(\neg f(\phi)) = 1$, namely $v_q(f(\chi)) = 1$. Suppose $v_q(f(\chi)) = 1$. Then, similarly, $v_p(\chi) = 1$. Thus, $v_q(f(\chi)) = 1$ iff $v_p(\chi) = 1$. Suppose $v_p(\chi) = 0$. Then $v_p(\phi) = 1$; hence, by IH, $v_q(f(\phi)) = 1$; and hence $v_q(\neg f(\phi)) = 0$, namely $v_q(f(\chi)) = 0$. Suppose $v_p(f(\chi)) = 0$. Then, similarly, $v_q(\chi) = 0$. Thus, $v_q(f(\chi)) = 0$ iff $v_p(\chi) = 0$.

(2) χ is $\phi \to \psi$, so $f(\chi) = f(\phi) \to f(\psi)$. Suppose $v_p(\chi) = 1$. Then $v_p(\phi) = 0$ or $v_p(\psi) = 1$; hence, by IH, $v_q(f(\phi)) = 0$ or $v_q(f(\psi)) = 1$; and hence $v_q(f(\phi) \to f(\psi)) = 1$, namely $v_q(f(\chi)) = 1$. Suppose $v_q(f(\chi)) = 1$. Then, similarly, $v_p(\chi) = 1$. Thus, $v_q(f(\chi)) = 1$ iff $v_p(\chi) = 1$. Similarly, $v_q(f(\chi)) = 0$ iff $v_p(\chi) = 0$. (The cases of other connectives are similar to this one.)

(3) χ is $\exists x\phi[x/c]$ (c occurs in ϕ and x is new to ϕ), so $f(\chi) = \exists T_x T \wedge f(\phi)[x/c]$. Suppose $v_p(\chi) = 1$. Then $v_p(\phi[d/c]) = 1$ for some d in L_p;

hence, by IH, $v_q(f(\phi[d/c])) = 1$ for some d in L_q; hence, by Proposition 6.3.3, $v_q(f(\phi)[d/c]) = 1$ for some d in L_q; hence $v_q(dT) = 1$ and $v_q(dT \wedge f(\phi)[d/c]) = 1$ for some d in L_q; and hence $v_q(f(\chi)) = 1$. Suppose $v_q(f(\chi)) = 1$. Then $v_q(dT) = 1$ and $v_q(dT \wedge f(\phi)[d/c]) = 1$ for some d in L_q; hence $v_q(f(\phi)[d/c]) = 1$ for some d in L_q; hence by Proposition 6.3.3 $v_q(f(\phi[d/c])) = 1$ for some d in L_q; hence by IH $v_p(\phi[d/c]) = 1$ for some d in L_p; and hence $v_p(\exists x\phi[x/c]) = 1$. Thus, $v_q(f(\chi)) = 1$ iff $v_p(\chi) = 1$. Similarly, $v_q(f(\chi)) = 0$ iff $v_p(\chi) = 0$.

(4) χ is $\forall x\phi[x/c]$ (c occurs in ϕ and x is new to ϕ), so $f(\chi) = \forall T_x T \wedge f(\phi)[x/c]$. Suppose $v_p(\chi) = 1$. Then $v_p(\phi[d/c]) = 1$ for every d in L_p; hence, by IH, $v_q(f(\phi[d/c])) = 1$ for every d in L_q; hence by Proposition 6.3.3 $v_q(f(\phi)[d/c]) = 1$ for every d in L_q; hence $v_q(dT) = 1$ and $v_q(dT \wedge f(\phi)[d/c]) = 1$ for every d in L_q; and hence $v_q(f(\chi)) = 1$. Suppose $v_q(f(\chi)) = 1$. Then $v_q(dT \wedge f(\phi)[d/c]) = 1$ for every d in L_q; hence $v_q(f(\phi)[d/c]) = 1$ for every d in L_q; hence by Proposition 6.3.3 $v_q(f(\phi[d/c])) = 1$ for every d in L_q; hence by IH $v_p(\phi[d/c]) = 1$ for every d in L_p; and hence $v_p(\chi) = 1$. Thus, $v_q(f(\chi)) = 1$ iff $v_p(\chi) = 1$. Similarly, $v_q(f(\chi)) = 0$ iff $v_p(\chi) = 0$.

■

6.5. Validity Preservation

PROPOSITION 6.5.1. *Let Γ be a set of formulas of a PC language L_p. We write $f(\Gamma)$ for the set of the translations of all the members of Γ. Let ϕ be a formula of L_p and $f(\phi)$ its translation. Then: $\Gamma \vDash_2 \phi$ iff $f(\Gamma) \vDash_3 f(\phi)$.*

PROOF. Let L_q be the Quarc language correlated with L_p. Then $f(\Gamma)$ is a set of L_q-formulas and $f(\phi)$ is an L_q-formula.

Suppose $f(\Gamma) \nvDash_3 f(\phi)$. Then, by the definition of validity, for some Quarc language L'_q there is a valuation v'_q on which all formulas in $f(\Gamma)$ are true and $f(\phi)$ is false. Let L'_p be the PC language with which L'_q is correlated, and v'_p the PC valuation (for L'_p) with which v'_q is correlated. Then, by Proposition 6.4.2, every formula in Γ is true on v'_p, and ϕ is false on v'_p. Hence, $\Gamma \nvDash_2 \phi$.

Suppose $\Gamma \nvDash_2 \phi$. Then, by the definition of validity, for some PC language L'_p there is a valuation v'_p on which all formulas in Γ are true and ϕ is false. Let L'_q be the Quarc language correlated with L'_p, and v'_q the Quarc valuation (for L'_q) correlated with v'_p. Then, by Proposition 6.4.2, every formula in $f(\Gamma)$ is true on v'_q, and $f(\phi)$ is false on v'_q. Hence, $f(\Gamma) \nvDash_3 f(\phi)$. ■

6.6. Soundness and Completeness in Quarc with T

We next consider the soundness and completeness of Quarc enriched with T. As we shall see, there are important differences between the two- and three-valued systems.

As for the proof system, we add to Definition 4.1.2 the derivation rule ThI (*Thing Introduction*), as follows:

– (i) aT ThI

where a is any singular argument.

The soundness preservation of ThI should be straightforward, since aT is true for any a on any valuation, on both the two- and three-valued systems. Given the soundness of the two-valued Quarc without T, it follows that the two-valued Quarc with T is also sound.

However, the soundness of the three-valued Quarc followed from the coincidence theorem above. Yet that proof does not apply to Quarc with T: given a 3-valuation w for a language, to generate a 2-valuation v for an enriched language such that if $w(\phi) = 1(0)$ then $v(\phi) = 1(0)$, it introduced a singular argument e such that, for any nonempty unary predicate P, $v(eP) = 0$. Since T is nonempty (i.e., for some a, $w(aT) = 1$), this would make $v(eT) = 0$, which would violate the semantic rule for *Thing*, namely, that for any singular argument a, $v(aT) = 1$. As we shall soon see, the proof system introduced above and enriched with ThI is unsound with respect to the three-valued Quarc with T.

The proof for the completeness of two-valued Quarc with T is essentially the same as in Sect. 5, noticing that any maximal consistent set of a language contains aT for any singular argument a in the language:

PROPOSITION 6.6.1. *Let Δ be a maximal consistent set of L-formulas. Then $aT \in \Delta$ for any a in L.*

PROOF. Suppose a is a singular argument in L. Then, by ThI, $\Delta \vdash aT$; and hence, by Proposition 5.2.2, $aT \in \Delta$. ∎

And since aT is a basic formula, the valuation induced by a maximal consistent set (the one employed in proving the Truth Lemma) complies with the valuation rule 'Thing'. Accordingly, the two-valued Quarc with T is sound and complete.

Consider now the formula, $\exists TP$. On the two-valued Quarc, due to Instantiation, it is true on any valuation, so $\vDash_2 \exists TP$. Since two-valued Quarc with T is complete, it follows that $\vdash \exists TP$, which is also simple to show directly. By contrast, since on the three-valued Quarc with T, while aT is true for

any a on any valuation, aP can be false for every a on a valuation, $\nvDash_3 \exists TP$. It follows, first, that $\vDash_2$ and $\vDash_3$ do not coincide on Quarc with T, and secondly, that the proof system developed above is unsound for three-valued Quarc with T.

Whether a sound and complete proof system for three-valued Quarc with T *can* be developed, we leave an open question. Notice, however, that Cut will not be *admissible* in such a system: since $\exists PP$ cannot be false on a valuation, $\vDash_3 \exists PP$; since if $\exists PP$ is true on a valuation, so is $\exists TP$, $\exists PP \vDash_3 \exists TP$; but as explained above, $\nvDash_3 \exists TP$. If a proof system for Quarc with T is sound and complete, the following should therefore hold on it: $\vdash \exists PP$, $\exists PP \vdash \exists TP$, $\nvdash \exists TP$. This would necessitate modifying the derivation rules for connectives introduced above. These issues arise in the translating system, the three-valued Quarc with T, despite their inexistence in the translated system, PC.

Ever since the first works of [8,9], the literature in this area has focused on formal systems which preserve their set of valid arguments when their two-valued semantics is replaced by a three-valued one and validity is defined as strict-to-tolerant (ST). This coincidence holds for the Propositional Calculus, the Predicate Calculus, and as we proved in Sect. 3.3, for Quarc (without T). The exceptions found in the literature for which there is no such coincidence are cases of paradox, specifically of the Predicate Calculus augmented with a truth predicate, or with a similarity predicate in the presence of vague concepts. While the system cannot then have a consistent 2-valuation because of, for instance, paradoxes generated by liar sentences, that is not the case with 3-valuations; in that case, Kripke's fixed point construction shows that there are 3-valued consistent models [13]. However, as we have just seen, a different result obtains for Quarc with T: the system has consistent 2- and 3-valuations, neither is paradoxical, but their inference relations do not coincide, for while $\vDash_2 \exists TP$, $\nvDash_{ST} \exists TP$.[7]

7. Quarc-to-PC' Translation

While in the previous section, we translated PC into an extended version of Quarc, Quarc plus T, we shall here translate Quarc into an extended version of PC. We need to introduce gaps into the valuations of the Predicate Calculus, and for that purpose we add a quantifier symbol $\exists'$ to Definition 6.1.1

[7] We are indebted to Pablo Cobreros for correspondence on this point.

(Language), and, correspondingly, in Definition 6.1.2 we add the following formation rule:

> If ϕ is a formula containing one or more occurrences of a, and x is a variable new to ϕ, then $\exists' x\phi[x/a]$ is a formula.

We will call this extended system **PC'**. The version of Quarc considered below is the three-valued one *without* T.

7.1. Three-valued TVS for PC'

Definition 7.1.1. (*3-valuation*) For a PC' language L, a *3-valuation* is a function v from L-formulas to $\{0, 1, u\}$ such that:

(a) For each basic formula ϕ of L, either $v(\phi) = 1$ or $v(\phi) = 0$.

(b) Let ϕ and ψ be formulas of L. Then:

$v(\neg\phi) = 1$ if $v(\phi) = 0$; $v(\neg\phi) = 0$ if $v(\phi) = 1$; otherwise $v(\phi) = u$.
$v(\phi \wedge \psi) = 1$ if $v(\phi) = 1$ and $v(\psi) = 1$; $v(\phi \wedge \psi) = 0$ if $v(\phi) = 0$ or $v(\psi) = 0$; otherwise $v(\phi \wedge \psi) = u$.
$v(\phi \vee \psi) = 1$ if $v(\phi) = 1$ or $v(\psi) = 1$; $v(\phi \vee \psi) = 0$ if $v(\phi) = 0$ and $v(\psi) = 0$; otherwise $v(\phi \vee \psi) = u$.
$v(\phi \rightarrow \psi) = 1$ if $v(\phi) = 0$ or $v(\psi) = 1$; $v(\phi \rightarrow \psi) = 0$ if $v(\phi) = 1$ and $v(\psi) = 0$; otherwise $v(\phi \rightarrow \psi) = u$.

(c) Let ϕ be a formula containing c, and x a variable new to ϕ. Then:

$v(\forall x\phi[x/c]) = 1$ if $v(\phi[d/c]) = 1$ for every d in L; $v(\forall x\phi[x/c]) = 0$ if $v(\phi[d/c]) = 0$ for some d in L; otherwise $v(\forall x\phi(x)) = u$.
$v(\exists x\phi[x/c]) = 1$ if $v(\phi[d/c]) = 1$ for some d in L; $v(\exists x\phi[x/c]) = 0$ if $v(\phi[d/c]) = 0$ for every d in L; otherwise $v(\exists x\phi[x/c]) = u$.
$v(\exists' x\phi[x/c]) = 1$ if $v(\phi[d/c]) = 1$ for some d in L; otherwise $v(\exists' x\phi[x/c]) = u$.

Definition 7.1.2. (*3-validity*) Let L be a PC' language. An argument whose premises constitute the set Γ of L-formulas and whose conclusion is the L-formula ϕ is *3-valid*, written $\Gamma \models_3 \phi$, iff for any PC' language which contains all the constants occurring in either Γ or ϕ, no 3-valuation assigns 'true' to all formulas in Γ and 'false' to ϕ.

7.2. The Translation Manual

Definition 7.2.1. (*Language correlation*) With each Quarc language L_q we correlate the unique PC' language L_p that satisfies the following requirements:

(a) The constants of L_p are the singular arguments of L_q.

(b) The predicates of L_p are those of L_q, and arities are preserved.

DEFINITION 7.2.2. (*Translation function*) The translation function t is defined recursively as follows:

(1) *Basic formulas:*
$t(a_1 \dots a_n P) = Pa_1 \dots a_n$

(2) *Reorder:*
$t(a_{\tau_1} \dots a_{\tau_n} P^\tau) = t(a_1 \dots a_n P)$

(3) *Negative predication:*
$t(a_1 \dots a_n \neg P) = \neg t(a_1 \dots a_n P)$

(4) *Truth functional compounds:*
$t(\neg\phi) = \neg t(\phi)$
$t((\phi * \psi)) = (t(\phi) * t(\psi))$, where $*$ is $\wedge$, $\vee$, or $\rightarrow$.

(5) *Anaphora:* Let $\phi(a_x)$ be a formula led by a labelled argument a_x, and ϕ the formula from which $\phi(a_x)$ is immediately generated. Then
$t(\phi(a_x)) = t(\phi)$

(6) *Quantification:* Let $\phi(QP)$ be a formula governed by an occurrence of QP, and ϕ the result of replacing the governing occurrence by c, a singular argument new to $\phi(QP)$. Let x be a variable new to $t(\phi)$. Then
$t(\phi(\forall P)) = \exists' x(Px) \wedge \forall x(Px \rightarrow t(\phi)[x/c])$
$t(\phi(\exists P)) = \exists' x(Px) \rightarrow \exists x(Px \wedge t(\phi)[x/c])$

The following proposition is easy to prove and will be used in later proofs.

PROPOSITION 7.2.3. *Let ϕ be a Quarc formula and $t(\phi)[d/c]$ the PC' formula which results from replacing every occurrence of c in $t(\phi)$ with d. Then:* $t(\phi)[d/c] = t(\phi[d/c])$.

7.3. Truth Value Preservation

We move on to show that the translation is adequate, in the sense that there is a bijection between Quarc valuations and PC' valuations, such that the truth values of Quarc formulas are preserved by the translation under this bijection. It will follow that the (in)validity of Quarc arguments is also preserved by the translation.

DEFINITION 7.3.1. (*Valuation correlation*) With each Quarc valuation v_q for a Quarc language L_q we correlate a PC' valuation v_p for L_p, the PC'

language that we correlate with L_q, such that: for every *basic* formula ϕ of Quarc, $v_p(t(\phi)) = v_q(\phi)$.

The following proof shows that, with valuations correlated, the Quarc-to-PC' translation preserves truth values.

PROPOSITION 7.3.2. *Let χ be a formula of a Quarc language L_q and $t(\chi)$ its translation in L_p, the PC' language that we correlate with L_q. Let v_q be a valuation for L_q and v_p the correlated valuation for L_p. Then: $v_p(t(\chi)) = v_q(\chi)$.*

PROOF. By induction on the complexity of L_q-formulas.

Base case: Suppose χ is a basic formula of L_q. Then by the valuation correlation $v_p(t(\chi)) = v_q(\chi)$.

Induction step: Assuming that the claim holds for all formulas less complex than χ, we consider the following cases:

(1) χ is a reordered form, $a_{\tau_1} \dots a_{\tau_n} P^{\tau}$, of a basic formula $a_1 \dots a_n P$, so $t(\chi) = t(a_1 \dots a_n P)$. Suppose $v_q(\chi) = 1$. Then $v_q(a_1 \dots a_n P) = 1$, hence by IH $v_p(t(a_1 \dots a_n P)) = 1$, and hence $v_p(t(\chi)) = 1$. Suppose $v_q(\chi) = 0$. Then, similarly, $v_p(t(\chi)) = 0$. As there is no *undefined* case for reordered forms of basic formulas, we can conclude that $v_p(t(\chi)) = v_q(\chi)$. (The case of negative predication is similar to this one.)

(2) χ is $\neg\phi$, so $t(\chi) = \neg t(\phi)$. Suppose $v_q(\chi) = 1$. Then $v_q(\phi) = 0$; hence, by IH, $v_p(t(\phi)) = 0$; and hence $v_p(\neg t(\phi)) = 1$, i.e. $v_p(t(\chi)) = 1$. Suppose $v_q(\chi) = 0$. Then, similarly, $v_p(t(\chi)) = 0$. Suppose $v_q(\chi) = u$. Then $v_q(\phi) = u$; hence, by IH, $v_p(t(\phi)) = u$; and hence $v_p(\neg t(\phi)) = u$, i.e. $v_p(t(\chi)) = u$. Hence, $v_p(t(\chi)) = v_q(\chi)$.

(3) χ is $\phi \to \psi$, so $t(\chi) = t(\phi) \to t(\psi)$. Suppose $v_q(\chi) = 1$. Then $v_q(\phi) = 0$ or $v_q(\psi) = 1$; hence, by IH, $v_p(t(\phi)) = 0$ or $v_p(t(\psi)) = 1$; and hence $v_p(t(\phi) \to t(\psi)) = 1$, i.e. $v_p(t(\chi)) = 1$. Suppose $v_q(\chi) = 0$. Then similarly $v_p(t(\chi)) = 0$. Suppose $v_q(\chi) = u$. Then similarly $v_p(t(\chi)) = u$. Hence, $v_p(t(\chi)) = v_q(\chi)$. (The cases of other connectives are similar to this one.)

(4) χ is led by a labelled singular argument. We write ϕ for the formula from which χ is immediately generated, so $t(\chi) = t(\phi)$. Suppose $v_q(\chi) = 1$. Then $v_q(\phi) = 1$; hence, by IH, $v_p(t(\phi)) = 1$; and hence $v_p(t(\chi)) = 1$. Suppose $v_q(\chi) = 0$. Then similarly $v_p(t(\chi)) = 0$. Suppose $v_q(\chi) = u$. Then similarly $v_p(t(\chi)) = u$. Hence, $v_p(t(\chi)) = v_q(\chi)$.

(5) χ is governed by an occurrence of $\exists P$. We write ϕ for the result of replacing the governing occurrence by c, a singular argument new to χ, so $t(\chi) = \exists' xPx \rightarrow \exists x(Px \wedge t(\phi)[x/c])$, where x is new to $t(\phi)$.

(5a) Suppose $v_q(\chi) = 1$. Then $v_q(dP) = 1$ and $v_q(\phi[d/c]) = 1$ for some d in L_q; hence, by IH and Proposition 7.2.3, $v_p(Pd) = 1$ and $v_p(t(\phi)[d/c]) = 1$ for some d in L_p; hence $v_p(\exists' xPx) = 1$ and $v_p(\exists x(Px \wedge t(\phi)[x/c])) = 1$; and hence $v_p(t(\chi)) = 1$.

(5b) Suppose $v_q(\chi) = 0$. Then similarly $v_p(t(\chi)) = 0$.

(5c) Suppose $v_q(\chi) = u$. Then we have the following two subcases:

(i) Suppose $v_q(aP) = 0$ for any singular argument a in L_q. Then $v_p(Pa) = 0$ for any constant a in L_p, hence $v_p(\exists' xPx) = u$ and $v_p(\exists x(Px \wedge t(\phi)[x/c])) = 0$, and hence $v_p(t(\chi)) = u$. (ii) Suppose $v_q(dP) = 1$ for some singular argument d in L_q, $v_q(\phi[d/c]) = u$ for some such d, and that $v_q(\phi[d/c]) \neq 1$ for every such d. Then, by IH and Proposition 7.2.3, $v_p(Pd) = 1$ for some d in L_p, $v_p(t(\phi)[d/c]) = u$ for some such d, and $v_p(t(\phi)[d/c]) \neq 1$ for every such d; hence $v_p(\exists' xPx) = 1$ and $v_p(\exists x(Px \wedge t(\phi)[x/c])) = u$; and hence $v_p(t(\chi)) = u$. Hence, $v_p(t(\chi)) = v_q(\chi)$. (The case of formulas governed by a universally quantified argument is similar to this one.) ■

7.4. Validity Preservation

PROPOSITION 7.4.1. *Let* Γ *be a set of formulas of a Quarc language* L_q *and we write* $t(\Gamma)$ *for the set of the translations of all the members of* Γ. *Let* ϕ *be a formula of* L_q *and* $t(\phi)$ *its translation. Then:* $\Gamma \vDash_3 \phi$ *iff* $t(\Gamma) \vDash_3 t(\phi)$.

PROOF. Let L_p be the PC' language correlated with L_q. Then $t(\Gamma)$ is a set of L_p-formulas and $t(\phi)$ is an L_p-formula.

Suppose $t(\Gamma) \nvDash_3 t(\phi)$. Then, by the definition of validity, for some PC' language L'_p there is a valuation v'_p on which every member of $t(\Gamma)$ is true and $t(\phi)$ is false. Let L'_q be the Quarc language with which L'_p is correlated, and v'_q the Quarc valuation (for L'_q) with which v'_p is correlated. Then, by Proposition 7.3.2, every member of Γ is true on v'_q, and ϕ is false on v'_q. Hence, $\Gamma \nvDash_3 \phi$.

Suppose $\Gamma \nvDash_3 \phi$. Then, by the definition of validity, for some Quarc language L'_q there is a valuation v'_q on which every member of Γ is true and ϕ is false. Let L'_p be the PC' language correlated with L'_q, and v'_p the PC' valuation (for L'_p) correlated with v'_q. Then, by Proposition 7.3.2, every member of $t(\Gamma)$ is true on v'_p, and $t(\phi)$ is false on v'_p. Hence, $t(\Gamma) \nvDash_3 t(\phi)$. ■

7.5. Provability in PC'

Similar issues to those of provability in the three-valued Quarc with T arise also for PC'. Let ϕ be a PC' formula containing a, and x a variable new to ϕ. Then, since in no case can $\exists' x\phi[x/a]$ be false, a PC' argument whose conclusion is $\exists' x\phi[x/a]$ is guaranteed to be valid. Hence:

$$\neg\exists x\phi[x/a] \vDash_3 \exists' x\phi[x/a]$$

And since whenever $\exists' x\phi[x/a]$ is true, $\exists x\phi[x/a]$ is true as well, we also have:

$$\exists' x\phi[x/a] \vDash_3 \exists x\phi[x/a]$$

Now, suppose that PC' has a proof system which is sound and complete. Then:

$$\neg\exists x(\phi[x/a]) \vdash \exists' x(\phi[x/a]) \text{ and } \exists' x(\phi[x/a]) \vdash \exists x(\phi[x/a])$$

However, $\neg\exists x(\phi[x/a]) \nvDash_3 \exists x(\phi[x/a])$. Accordingly, Cut is not admissible in this proof system, otherwise we would have $\neg\exists x(\phi[x/a]) \vdash \exists x(\phi[x/a])$. This despite the fact that no such issue exists for the translated system, three-valued Quarc.

8. Conclusion

In this paper, we developed both a two-valued and three-valued truth-valuational semantics for the Quantified Argument Calculus (Quarc). The two-valued version followed closely that in [2], which included an Instantiation rule, forcing unary predicates to have instances. The three-valued semantics eliminated this rule, taking formulas governed by either $\forall P$ or $\exists P$ to *presuppose* that P has instances and making them truth-value-less otherwise. The elimination of the Instantiation rule in this way creates a richer and in a sense a semantically more natural system, which is therefore of much interest. This approach was also followed in [15], but unlike that paper, this one adopted strict-to-tolerant and not strict-to-strict validity. We then proved a coincidence result: $\Gamma \vDash_2 \phi$ iff $\Gamma \vDash_3 \phi$. This result *does not* hold on the SS validity approach. For example, since $\exists PP$ is never false, $\vDash_2 \exists PP$ and $\vDash_{ST} \exists PP$, but $\nvDash_{SS} \exists PP$.

We then provided a Natural Deduction proof system and proved completeness for the two-valued Quarc. That of the three-valued version followed from the coincidence theorem. Although our proof, using Lindenbaum's Lemma, is close in method to the one used in [18, §2.3], it was

hopefully of some additional interest, as the adaptation of such proofs to truth-valuational semantics is not found in later literature. We believe these results contribute to the interest in the three-valued Quarc with ST validity.

Lastly, we investigated the relations between the three-valued Quarc and PC. (The relation of Quarc as a *two*-valued system to PC has been addressed in several works [16,25,29].) The unrestricted nature of PC quantification was imitated in Quarc by adding to it a logical predicate T, which applies to all singular arguments of all languages. The gappy nature of three-valued Quarc quantifiers was imitated in PC by adding to it a weak existential quantifier, $\exists'$, for which $\exists' x\phi(x)$ is truth-value-less in case it has no instances. We managed in this way to incorporate a semantic image of each calculus in the extension of the other, in the sense that an argument in the one is valid just in case so is its translation into the other. It should be emphasised that neither extension is justified by internal considerations on its system (unlike the elimination of Instantiation), but was done only in order to imitate features of the other system. In fact, as we saw, either extension creates difficulties for a proof system for its calculus, forcing it not to admit Cut if sound and complete, difficulties inexistent either in the unextended version or in the translated calculus. Although we have not proved that a different, less problematic extension of either system which allows for an incorporation as above is impossible, other options we have tried had similar drawbacks and so far none has been suggested in the literature. To this extent, our results support a claim already made in several works, that although related, Quarc and PC are essentially different calculi.

Funding Open access funding provided by Central European University Private University

References

[1] Ben-Yami, Hanoch, *Logic & Natural Language: On Plural Reference and Its Semantic and Logical Significance*. London, Routledge (2004). https://doi.org/10.4324/9781315250267.

[2] Ben-Yami, Hanoch, The quantified argument calculus, *The Review of Symbolic Logic*, 7(1):120–146 (2014). https://doi.org/10.1017/S1755020313000373.

[3] Ben-Yami, Hanoch, The quantified argument calculus and natural logic, *Dialectica* 74(2):35–70 (2020). https://doi.org/10.48106/dial.v74.i2.02.

[4] Ben-Yami, Hanoch, The Barcan formulas and necessary existence: the view from Quarc, *Synthese* 198(11):11029–10164 (2021). https://doi.org/10.1007/s11229-020-02771-4.

[5] Ben-Yami, Hanoch. manuscript, Truth and Proof without Models: A Development and Justification of the Truth-Valuational Approach.

[6] Ben-Yami, Hanoch, and Edi Pavlović, Forthcoming, Completeness of the Quantified Argument Calculus on the Truth-Valuational Approach, in Boran Berčić, Aleksandra Golubović, and Majda Trobok, (eds.), *Human Rationality: Festschrift for Nenad Smokrović, Faculty of Humanities and Social Sciences*, University of Rijeka, pp. 53–67.

[7] Chiswell, Ian, and Wilfrid Hodges. *Mathematical Logic*, Oxford Texts in Logic. Oxford, New York: Oxford University Press (2007).

[8] Cobreros, Pablo, Paul Egré, David Ripley, and Robert van Rooij, Tolerant, classical, strict, *Journal of Philosophical Logic* 41(2):347–385 (2012). https://doi.org/10.1007/s10992-010-9165-z.

[9] Cobreros, Pablo, Paul Egré, David Ripley, and Robert van Rooij, Reaching transparent truth, *Mind* 122(488):841–866 (2013). https://doi.org/10.1093/mind/fzt110.

[10] Cobreros, Pablo, Paul Egré, David Ripley, and Robert van Rooij, Vagueness, Truth and Permissive Consequence, in Theodora Achourioti, Henri Galinon, José Martínez Fernáindez, and Kentaro Fujimoto (eds.),]*Unifying the Philosophy of Truth. Logic, Epistemology and the Unity of Science*, vol. 36, Dordrecht, Springer Netherlands, pp. 409–430.

[11] Dunn, J. Michael, and Nuel D. Belnap. 1968, The substitution interpretation of the quantifiers, *Noûs* 2(2):177–185. https://doi.org/10.2307/2214704.

[12] Halldén, Sören. *The Logic of Nonsense*. Uppsala (1949).

[13] Kripke, Saul A, Outline of a Theory of Truth. Reprinted in his 2011. *Philosophical Troubles: Collected Papers*, I. Oxford: Oxford University Press, pp. 75–98 (1975).

[14] Kripke, Saul A, Is There a Problem About Substitutional Quantification? In *Truth and Meaning*, edited by Gareth Evans and John McDowell, 325–419. Oxford University Press (1976).

[15] Lanzet, Ran, A three-valued quantified argument calculus: Domain-free model-theory, completeness, and embedding of FOL, *The Review of Symbolic Logic* 10(3):549–582 (2017). https://doi.org/10.1017/S1755020317000053.

[16] Lanzet, Ran, and Hanoch Ben-Yami. 2004, Logical Inquiries into a New Formal System with Plural Reference, in Vincent Hendricks, Fabian Neuhaus, Stig Pedersen,

Uwe Scheffler, and Heinrich Wansing (eds.), *First-Order Logic Revisited*, 173–223. Berlin: Logos Verlag.

[17] Leblanc, Hugues, A simplified account of validity and implication for quantificational logic, *The Journal of Symbolic Logic* 33(2):231–235 (1968). https://doi.org/10.2307/2269868.

[18] Leblanc, Hugues, *Truth-Value Semantics*. Amsterdam, New York, and Oxford: North-Holland Publishing Company (1976).

[19] Leblanc, Hugues, Alternatives to Standard First-Order Semantics, in D. Gabbay and F. Guenthner(eds.), *Handbook of Philosophical Logic: Volume I: Elements of Classical Logic*, Synthese Library, Dordrecht, Springer Netherlands, pp. 189–274 (1983). https://doi.org/10.1007/978-94-009-7066-3_3.

[20] Lewis, H. A., Substitutional Quantification and Nonstandard Quantifiers, *Noûs* 19(3): 447–451 (1985). https://doi.org/10.2307/2214953.

[21] Pascucci, Matteo, An Axiomatic Approach to the Quantified Argument Calculus. *Erkenntnis*, January (2022). https://doi.org/10.1007/s10670-022-00519-9.

[22] Pavlović, Edi, *The Quantified Argument Calculus: An Inquiry into Its Logical Properties and Applications*. PhD thesis, Central European University, Budapest (2017).

[23] Pavlović, Edi, and Norbert Gratzl, Free Logic and the Quantified Argument Calculus, in Gabriele M. Mras, Paul Weingartner, and Bernhard Ritter(eds.), *Philosophy of Logic and Mathematics: Proceedings of the 41st International Ludwig Wittgenstein Symposium*, Berlin, Boston: De Gruyter (pp. 105–116)(2019a). https://doi.org/10.1515/9783110657883-007.

[24] Pavlović, Edi, and Norbert Gratzl, Proof-Theoretic Analysis of The Quantified Argument Calculus. *The Review of Symbolic Logic* 12(4):607–636 (2019b). https://doi.org/10.1017/S1755020318000114.

[25] Pavlović, Edi, and Norbert Gratzl, Abstract Forms of Quantification in the Quantified Argument Calculus. *The Review of Symbolic Logic*, March, 1-31 (2021). https://doi.org/10.1017/S175502032100006X.

[26] Peano, Giuseppe, *Formulaire de Mathématique*. Turin: Bocca Frères, CH. Clausen (1897).

[27] Priest, Graham, The logic of paradox, *Journal of Philosophical Logic* 8(1):219–241 (1979). https://doi.org/10.1007/BF00258428.

[28] Priest, Graham, *Towards Non-Being*. Second Edition. Oxford, New York: Oxford University Press (2016).

[29] Raab, Jonas, The Relationship of QUARC and Classical Logic. München: Ludwig-Maximilians-Universität München (2016).

[30] Raab, Jonas, Aristotle, Logic, and QUARC, *History and Philosophy of Logic* 39(4):305–340 (2018). https://doi.org/10.1080/01445340.2018.1467198.

[31] Strawson, P. F., On Referring, *Mind* 59(235):320–344 (1950).

[32] Strawson, P. F., *Introduction to Logical Theory*. London: Methuen (1952).

[33] Strawson, P. F., Identifying reference and truth-values, *Theoria* 30(2):96–118 (1964). https://doi.org/10.1111/j.1755-2567.1964.tb00404.x.

[34] Westerståhl, Dag. manuscript, *Foundations of First-Order Logic: Completeness, Incompleteness, Computability*.

H. Yin, H. Ben-Yami
Central European University
Vienna
Austria
benyamih@ceu.edu