THE REVIEW OF SYMBOLIC LOGIC
Volume 10, Number 3, September 2017

A THREE-VALUED QUANTIFIED ARGUMENT CALCULUS: DOMAIN-FREE MODEL-THEORY, COMPLETENESS, AND EMBEDDING OF FOL

RAN LANZET
Tel-Aviv University and the Hebrew University of Jerusalem

Abstract. This paper presents an extended version of the Quantified Argument Calculus (Quarc). Quarc is a logic comparable to the first-order predicate calculus. It employs several nonstandard syntactic and semantic devices, which bring it closer to natural language in several respects. Most notably, quantifiers in this logic are attached to one-place predicates; the resulting quantified constructions are then allowed to occupy the argument places of predicates. The version presented here is capable of straightforwardly translating natural-language sentences involving defining clauses. A three-valued, model-theoretic semantics for Quarc is presented. Interpretations in this semantics are not equipped with domains of quantification: they are just interpretation *functions*. This reflects the analysis of natural-language quantification on which Quarc is based. A proof system is presented, and a completeness result is obtained. The logic presented here is capable of straightforward translation of the classical first-order predicate calculus, the translation preserving truth values as well as entailment. The first-order predicate calculus and its devices of quantification can be seen as resulting from Quarc on certain semantic and syntactic restrictions, akin to simplifying assumptions. An analogous, straightforward translation of Quarc into the first-order predicate calculus is impossible.

§1. Introduction. Interest in extensional logics whose quantificational devices are closer to those of natural languages than are the ones of the standard first-order predicate calculus has existed for some time (see, e.g., Barwise & Cooper, 1981; Ben-Yami, 2014; Francez, 2014, see Francez 2014 for further references). Ben-Yami (2014) presents a *quantified argument calculus* (Quarc).[1] An earlier version of this logic is presented in (Lanzet & Ben-Yami, 2004). The quantified argument calculus is a logic in which quantifiers attach to one-place predicates rather than to open formulas; the resulting quantified constructions, expressions of the forms $\forall P$ and $\exists P$, where P is a one-place predicate, are allowed to occupy the argument places of predicates. This is similar to what we find in natural languages: consider the syntactic positions occupied by the expressions 'every professor' and 'some courses' in 'Every professor teaches some courses'.

The logic of (Ben-Yami, 2014) is arguably closer to natural languages, in several syntactic and semantic respects, than any of the various versions of the predicate calculus. There are, however, three points on which Ben-Yami's treatment of Quarc can be improved: first,

Received: July 24, 2016.
2010 *Mathematics Subject Classification*: 03B65, 03A05, 03B50, 03B60.
Key words and phrases: quantified argument calculus, Quarc, three-valued logic, three-valued Quarc, 3-Quarc, quantified argument, natural language, anaphora, defining clause, domain, domain of quantification, universe, domain-free model theory, completeness, reordered form, Square of Opposition.

[1] He also compares this logic with related logical systems, such as the ones of (Pratt-Hartmann & Moss, 2009), (Moss, 2010), and (Francez, 2014).

doi:10.1017/S1755020317000053

his semantics relies on what I will call a *global nonemptiness requirement*: all one-place predicates are required, as part of the semantic framework, to have instances.[2] Counter-intuitive consequences of this requirement are mentioned in §2 below. Second, Ben-Yami does not incorporate into the logic a treatment of defining clauses, which often modify predicates in natural languages. Consequently, while sentences such as

Every person is rich (1)

can be straightforwardly formalized in his logic, there is no straightforward way to formalize sentences such as

Every person *who owns a private jet* is rich (2)

(More on this in §2). Third, although Ben-Yami writes that he intends to provide in a different paper a proof of the completeness of the system he develops, such a proof has not yet been provided.

In this paper, an extended and improved version of Quarc is developed. (Unless otherwise indicated, the term 'Quarc' will henceforth be used for the version presented here.) The logic presented in this paper has the syntactic and semantic resources to treat defining clauses, and can straightforwardly translate sentences such as (2). The semantics presented here is model-theoretic and three-valued (unlike that of (Ben-Yami, 2014), which is truth-valuational and bivalent). A proof-system is formulated, and a completeness result is obtained. As explained below, the three-valued semantic framework adopted here makes it possible to dispense with the global nonemptiness requirement mentioned above.

The notion of interpretation employed here differs from the standard one: in Quarc semantics, interpretations are not equipped with a domain of quantification; rather, they are just interpretation *functions*. As will be seen below, this feature of Quarc reflects the analysis of natural-language quantification on which this logic is based. It will also be seen that the development of a domain-free semantics for Quarc does not require any special technical effort: the need for domains of quantification in Quarc simply does not arise. More on this will be said in §2, §4 and §8.

The devices allowing the translation of defining clauses into Quarc will also enable a straightforward translation of the sentences of standard first-order predicate calculus (PC). These will be translated into Quarc in much the same way as they are ordinarily translated into English. The translation will be shown to preserve truth values, as well as entailment. Quarc will thus be seen to offer not only formalization options unavailable in PC, but also a PC-style formalization of whatever can be formalized in PC. It will also be shown that, while an entailment-preserving translation of Quarc into PC is possible, no such translation can be straightforward. The relevant notion of a straightforward translation—on which PC is straightforwardly translatable into Quarc but not vice versa—will be explained in §7.

§2 contains a semi-formal presentation of Quarc. This presentation covers both the aspects in which the current version of Quarc differs from that of (Ben-Yami, 2014) and the ones in which the two versions overlap. The latter are covered in order to make the presentation self-contained. §3 and §4 contain the exact definitions of Quarc: syntactic definitions in the former, and semantic ones in the latter. §5 contains a proof system for Quarc; and §6—a proof of the completeness of this system. §7 treats the translation of PC-sentences into Quarc and vice versa. In §8 it is explained how PC can be seen as

[2] The same principle was relied on in (Lanzet & Ben-Yami, 2004).

resulting from Quarc on certain restrictions, akin to simplifying assumptions. The way in which quantification is achieved in PC—by attaching quantifiers to open formulas and by reference to a fixed, presupposed domain—emerges as a result of those restrictions rather than as a necessary consequence of treating quantification in logic and in model-theory.

§2. A semi-formal presentation. The analysis on which Quarc is based can be illustrated with respect to the following pair of sentences:[3]

All students are polite (1)

Some students are polite (2)

In each of these two sentences, a quantifier—'all' or 'some'—is attached to a noun—'students'. The role of this noun is to introduce the plurality required for the operation of the attached quantifier: it introduces all the students.[4] The role of the attached quantifier, in turn, is to determine to how many of the individuals introduced by 'students' the predicate 'polite' is applied. Both (1) and (2) involve, on the current analysis, the presupposition that the extension of 'students' is nonempty; if that extension *is* empty, then the plurality required for the operation of the attached quantifier is not introduced, and the sentence fails to express a true or false proposition.[5]

This analysis differs from the Frege-Russell analysis of sentences such as (1) and (2): unlike on the Frege-Russell account, we do not analyze 'students' in (1) and (2) as predicative; and we do not consider those sentences to refer to every element of some fixed domain, of which the extension of 'students' is a proper subclass. Sentence (1), for instance, does not state *of everything* that if it is a student then it is polite. Rather, the sentence asserts *of the students alone* that all of them are polite.[6]

The above analysis will be reflected in Quarc in the following way: first, nouns such as 'students' will be represented by one-place predicates, and quantifiers will attach to such predicates rather than to open formulas. (1) and (2) will be formalized in Quarc, respectively, as:[7]

$$(\forall S)\ P \tag{3}$$

$$(\exists S)\ P \tag{4}$$

The expressions $\forall S$ and $\exists S$, occupying the argument place of P, will be called *quantified arguments*, or QAs. One-place predicates will thus have two distinct uses in Quarc: they will be employed both in the ordinary, predicative role and as parts of QAs.

Second, the predicates to which quantifiers attach—such as S in (3) and (4)—will be interpreted as introducing the plurality over which we quantify. For this reason, we will

[3] The analysis is close to the ones of (Strawson, 1950) and (Geach, 1962). For a detailed version of it, see (Ben-Yami, 2004).

[4] More precisely: it introduces all the students *that are under consideration in the context in which the sentence is uttered*; but we will not try to capture such contextual features of sentence-utterances here.

[5] This is not meant as an analysis of presupposition in general.

[6] Cf. (Francez, 2014, 1164).

[7] I follow (Ben-Yami, 2014) in writing the arguments to the left of the predicate. This will help distinguishing the formulas of Quarc from those of PC in contexts in which both systems are discussed.

have no need for a fixed, presupposed plurality, in the form of a domain of quantification: interpretations in Quarc-semantics will be just interpretation *functions*, assigning sets of n-tuples to n-place predicates, and assigning any objects whatsoever to individual constants. Third, we will employ a three-valued framework, and let (3) and (4) be neither true nor false in interpretations $\mathcal{M}$ in which the extension $S^{\mathcal{M}}$ of S is empty.

Quarc will thus differ from PC in at least three respects: (i) the employment of quantified arguments; (ii) the domain-free notion of interpretation; and (iii) the three-valued semantics and the presuppositional approach (on which (3), for instance, is neither true nor false in interpretations in which the extension of S is empty). Moreover, (i)–(iii) can be said to be mutually-independent, in the following sense: we can decide, for each of them, whether to adopt it or not; and a logic can be developed based on each of the possible combinations of such decisions. I will make two comments here: first, I believe that in developing a logic that differs from PC in (i)–(iii), I will not be mixing up three unrelated issues. For the choice to adopt (i)–(iii) is not arbitrary: it is motivated by a single account of natural-language quantification. Second, pointing out the independence (in the above sense) of (i)–(iii) is not, by itself, an objection to Quarc any more than it is an objection to PC; for the latter, just like former, is based on one particular combination of decisions with respect to (i)–(iii).

Formally, (3) and (4) will be assigned a third value, U ('undefined') in interpretations in which the extension of S is empty. We will thus have the following, where $\varphi^{\mathcal{M}}$ is the value assigned to the formula φ in the interpretation $\mathcal{M}$:

$$[(\forall S)P]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } S^{\mathcal{M}} \neq \emptyset \text{ and } (x)P \text{ is true in } \mathcal{M} \text{ of every element of } S^{\mathcal{M}}, \\ \text{F} & \text{if } S^{\mathcal{M}} \neq \emptyset \text{ and } (x)P \text{ is false in } \mathcal{M} \text{ of some element of } S^{\mathcal{M}}, \\ \text{U} & \text{otherwise.} \end{cases}$$

$$[(\exists S)P]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } S^{\mathcal{M}} \neq \emptyset \text{ and } (x)P \text{ is true in } \mathcal{M} \text{ of some element of } S^{\mathcal{M}}, \\ \text{F} & \text{if } S^{\mathcal{M}} \neq \emptyset \text{ and } (x)P \text{ is false in } \mathcal{M} \text{ of every element of } S^{\mathcal{M}}, \\ \text{U} & \text{otherwise.} \end{cases}$$

Note that the need for a fixed domain of quantification does not arise here: the plurality over which we quantify is supplied by S.

As a consequence of this semantic treatment, (3)—the translation of 'All students are polite'—will entail (4)—'Some students are polite'. In this, Quarc is similar to Aristotelian logic, and diverges from PC, as well as from, e.g., the generalized-quantifier logic of (Barwise & Cooper, 1981).[8] The ordinary PC approach to this matter seems to fit the standard use of sentences of the relevant forms *in mathematics*. In ordinary contexts, however, the PC rendering of the relevant sentences seems highly counterintuitive, at least

[8] Cf. (Strawson, 1950, 343–344), where an analysis similar to the one relied on in this paper is put forward in defense of the Aristotelian Square of Opposition.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

when we limit ourselves to an extensional fragment of English (which is where PC is supposed to apply).[9]

The truth or falsity of

$$(x)P \qquad (\text{`}x \text{ is polite'}) \tag{5}$$

of a given object in a given interpretation will be defined in the following, straightforward way: (5) will be true of α in $\mathcal{M}$ iff

$$(c)P \tag{6}$$

is true in $\mathcal{M}^{\alpha}_{c}$, the interpretation resulting from $\mathcal{M}$ on replacing $c^{\mathcal{M}}$ (the denotation in $\mathcal{M}$ of c) with α. The falsity of (5) of α will be defined similarly.

Atomic, or *basic* formulas such as (6) will be treated classically: (6) will be true in an interpretation $\mathcal{M}$ if $c^{\mathcal{M}} \in P^{\mathcal{M}}$ and false otherwise. Basic formulas of the form

$$(c_1, \ldots, c_n)\, R, \tag{7}$$

where c_i are individual constants and R is an n-place predicate, will be treated similarly.

Truth-functional compounds will be constructed in the usual manner; semantically, they will be treated using Kleene's strong three-valued truth tables.[10] Despite the reliance on Kleene's truth tables, there will be a significant difference between Quarc and Kleene's strong three-valued logic: unlike in Kleene's logic, the (only) source of truth-value gaps in Quarc will be the empty extensions of one-place predicates occurring within QAs. As stated above, atomic, or basic, formulas will be treated classically in Quarc, and will not give rise to such gaps.

---

[9] Thus, for instance, few would take the sentence

All my children work in the coal mines (i)

as expressing a truth, when uttered by a childless person.

Other examples may seem to suggest that sentences of the forms 'All *A*s are *B*s', or 'Any *A* is *B*', do *not* presuppose the nonemptiness of the term *A*, and are thus better captured by their PC-formalizations. The following, discussed in (Kneale & Kneale, 1971, 61), is a typical example:

Any person who thinks he can swim the Atlantic is a fool. (ii)

Intuitively, this sentence is true even if no one thinks he can swim the Atlantic. This shows that (ii) lacks the relevant presupposition, and allegedly supports the standard first-order formalization of that sentence. On closer examination, however, the formalization of (ii) and similar sentences as $\forall x\,(Ax \rightarrow Bx)$ is totally inadequate. For consider the sentence

Any person who thinks he can swim the Atlantic *can* swim the Atlantic. (iii)

This sentence is intuitively false, even if no one thinks he can swim the Atlantic; and this is incompatible with the translation of (iii) in the form $\forall x\,(Ax \rightarrow Bx)$. What these examples in fact suggest is that the semantics of sentences such as (ii) and (iii)—universal sentences expressing a general law—is not extensional. If this is so, then, as long as we limit the discussion to an extensional fragment of English, examples such as (ii) and (iii) will be irrelevant.

[10] This is not to say that Kleene's strong truth-tables are the last word on presupposition projection. I do not claim here that they offer more than a rough model of how presuppositions are transferred, or projected, from truth-functional components to the relevant compounds. However, since the projection problem is not the focus of this paper, since Kleene's strong truth-tables have been employed in formal work on presupposition, and since they are relatively familiar and simpler than some other treatments of presupposition projection, I adopt them for the purposes of this work. Different versions of Quarc can, of course, be developed based on other approaches to presupposition.

It would have been possible, of course, to allow truth-value gaps already at the level of basic formulas, thus departing from classical PC at an additional point. The decision whether or not to do so is independent of the three-valued treatment of quantification described above. I chose not to allow this second source of truth-value gaps in Quarc in order to keep the focus on quantification, in order to avoid unnecessary deviations from classical PC (and thus, perhaps, to increase familiarity and facilitate an easier comparison between Quarc and PC), and in order to avoid unnecessary complication of the semantics.

Variables in Quarc will function somewhat differently than in PC: they will be employed as *anaphors*. An anaphor here is a expression whose basic function is to refer to whatever is referred to by its *source*: an individual constant occurring earlier in the same formula. To correlate variable occurrences with their sources, we will write the variable in subscript to the right of its source. We will thus have formulas such as:

$$(j_x, x)L \qquad \text{('John loves himself')}. \tag{8}$$

We will say that the second occurrence of x in (8) is an anaphor of (the occurrence of) j in that formula. Semantically, (8) will be treated as equivalent to

$$(j, j)L \qquad \text{('John loves John')}. \tag{9}$$

To illustrate the interaction of variables with quantifiers in Quarc, consider the sentence

$$\text{Every man loves himself.} \tag{10}$$

This sentence results from 'John loves himself' on replacing 'John' with 'every man'. The same will happen in Quarc: (10) will be represented by a formula resulting from (8) on replacing j with $\forall M$:

$$(\forall M_x, x)L. \tag{11}$$

The source of the second occurrence of x in this formula will be said to be the quantified argument $\forall M$.

Semantically, (11) will be treated along the lines already indicated in the case of (3) above:

$$[(\forall M_x, x)L]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } M^{\mathcal{M}} \neq \emptyset \text{ and } (y_x, x)L \text{ is true in } \mathcal{M} \text{ of every element of } M^{\mathcal{M}}, \\ \text{F} & \text{if } M^{\mathcal{M}} \neq \emptyset \text{ and } (y_x, x)L \text{ is false in } \mathcal{M} \text{ of some element of } M^{\mathcal{M}}, \\ \text{U} & \text{otherwise.} \end{cases}$$

Also, similarly to what we saw with (5), the expression $(y_x, x)L$ ('y loves himself') will be true of α in $\mathcal{M}$ iff (8) is true in $\mathcal{M}^{\alpha}_{j}$.

The truth conditions of quantified formulas such as (11) will thus in effect be determined on the basis of the truth conditions of their instances—formulas such as (8). This is similar to what we saw above in the case of $(\forall S)\, P$ (formula (3)) and its instances—formulas such as $(c)P$ (formula (6)). It can also be seen that, unlike in PC and related systems, variables enter the quantified formulas of Quarc only in cases where they occur already in the formula's instances: a variable is employed in (11), just as it was employed in (8); but no variable is employed in either the quantified (3) or its instance (6). Quarc is arguably closer to natural language on this point. For consider the English sentences represented by

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

(11), (8), (3), and (6): while the reflexive pronoun 'himself' is employed in both 'Every man loves himself' and 'John loves himself', no such device is employed in either 'Every student is polite' or 'John is polite'.

This interaction of quantifiers, variables, and individual constants will be reflected in the exact definitions presented in §3 and §4 below: variables, whose basic function is as anaphors of individual constants, will be introduced into formulas by a syntactic operation transforming, e.g., (9) into (8). Quantified formulas will be generated by another syntactic operation, replacing an occurrence of an individual constant—whether or not it has anaphors—by a QA. This second operation will take us from (8) to (11) and from (6) to (3).

The same principles illustrated above will be applied in cases where the variable (anaphor) and its source are separated by connectives. The sentences

If John loves Mary, then she loves him (12)

If a man loves Mary, then she loves him (13)

will be formalized, respectively, as

$$(j_x, m_y)L \rightarrow (y, x)L \tag{14}$$

$$(\forall M_x, m_y)L \rightarrow (y, x)L. \tag{15}$$

Note that the formalization of (13) as (15) involves rewriting 'a man' as the universally quantified $\forall M$. This is simlar to what we find in the PC formalization of such donkey sentences.[11]

Semantically, (15) will be treated by the principles already outlined above. We will thus have:

$$[(\forall M_x, m_y)L \rightarrow (y, x)L]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } M^{\mathcal{M}} \neq \emptyset \text{ and } (z_x, m_y)L \rightarrow (y, x)L \text{ is} \\ & \text{true in } \mathcal{M} \text{ of every element of } M^{\mathcal{M}}, \\ \text{F} & \text{if } M^{\mathcal{M}} \neq \emptyset \text{ and } (z_x, m_y)L \rightarrow (y, x)L \text{ is} \\ & \text{false in } \mathcal{M} \text{ of some element of } M^{\mathcal{M}}, \\ \text{U} & \text{otherwise.} \end{cases}$$

The expression $(z_x, m_y)L \rightarrow (y, x)L$ will be true of α in $\mathcal{M}$ if (14) is true in $\mathcal{M}^{\alpha}_{j}$; and (14) will be treated as equivalent to $(j, m)L \rightarrow (m, j)L$.

Multiply quantified sentences, such as

Every man loves some women, (16)

have been claimed to involve scope ambiguity. According to Ben-Yami (2014, 127), the prevalence of this ambiguity has been exaggerated; and even in cases in which ambiguity does exist, sentences of the form of (16) usually have a default reading, on which the first (leftmost) quantified phrase, ('every man', in the case of (16)), has wider scope than the second ('some women'). This default reading is rejected, according to Ben-Yami, only in case it does not make sense in the context. This analysis is reflected in the logical system of (Ben-Yami, 2014) by the adoption of a uniform principle for quantifier scope, or priority,

[11] This rewriting is avoided on dynamic treatments, such as those of DRT and DPL. Although I will not pursue this direction here, Quarc is sufficiently similar to PC to allow the development of a dynamic semantics along the lines of (Groenendijk & Stokhof, 1991).

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

designed to capture the default reading. This principle takes into account the order in which the QAs occur in the given formula, as well as other syntactic features.

I consider Ben-Yami's approach to multiply quantified sentences in natural language plausible. It also has the advantage of being able to explain the semantic role of the passive voice, as well as of converse relation expressions, in multiply quantified sentences: these function as reordering devices, enabling us to change the quantifiers' order of priority in a sentence (this is illustrated below). I will here follow Ben-Yami and develop Quarc around a uniform principle for determining the priority of QAs. Correspondingly, I will incorporate into Quarc reordering devices that imitate the ones found (on Ben-Yami's approach) in natural language.

An alternative approach would be that of (Francez, 2014): attachment of numerical indices to quantified expressions and determination of quantifier priority in accordance with those indices. Francez's approach has the advantage of enabling a straightforward representation of more than one reading of sentences of the form of (16) in cases that do involve scope ambiguity. A downside of this approach, however, is that the resulting logic no longer reflects the semantic role of natural language devices such as converse relation expressions (Ben-Yami, 2014).

A third formal approach, more flexible than either Ben-Yami's (and the one adopted here) or Francez's would be the following: (i) employ a uniform principle for quantifier priority based on the QAs' order of occurrence in the formula, and treat the reading of multiply quantified formulas resulting from this principle as a default reading; (ii) allow the attachment of numerical indices to QAs, and let the reading dictated by those indices override the default reading whenever such indices are present. If Ben-Yami's view of multiple quantification in natural language is correct, then this would bring Quarc even closer to natural language. I will not apply this approach here, in order not to overly complicate the syntax. It should be noted, however, that Quarc can be easily modified so as to incorporate this third approach.

Let us now present, in some detail, the way in which quantifier priority will be determined in Quarc. Consider again the sentence (16) ('Every man loves some women'). This will be represented in Quarc as

$$(\forall M, \exists W)\, L. \quad [12] \tag{17}$$

Quantifier priority in this formula will be determined in accordance with the order of QA occurrences. The main quantified argument of (17)—the one *governing* that formula—will be $\forall M$: the formula will be true in an interpretation $\mathcal{M}$ iff $M^{\mathcal{M}}$ is nonempty and

$$(x, \exists W)\, L \qquad (\text{'}x \text{ loves some women'})$$

is true in $\mathcal{M}$ of every element of $M^{\mathcal{M}}$.

The general principle determining quantifier-priority in Quarc will take into consideration more than just the order in which quantifiers occur in a given formula. I will explain

[12] Compare (17) with the formalization of (16) in the logic **L(GQ)** of (Barwise & Cooper, 1981):

$$\textbf{every}\ (\mathbf{M})\ \hat{x}\ \left[\textbf{some}\ (\mathbf{W})\ \hat{y}\ \left[\mathbf{L}\ (x, y)\right]\right]. \tag{i}$$

Although **L(GQ)** is syntactically closer to natural languages such as English than PC is, there are aspects in which Quarc is yet closer to natural languages. In the **L(GQ)** formalization (i), the quantifiers are applied from the outside to expressions resembling open formulas (Barwise and Cooper's 'set terms', such as $\hat{x}\ \left[\textbf{some}\ (\mathbf{W})\ \hat{y}\ \left[\mathbf{L}\ (x, y)\right]\right]$). Since this is how quantification works in **L(GQ)**, the need for variables arises already in examples such as (16). No similar syntactic device is employed in either the English (16) or the Quarc formalization (17).

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

this principle using the following example:

If all forum members complain about a post, then it gets deleted. (18)

In PC this becomes:

$$\forall x \left(\left[Px \wedge \forall y \left(Fy \rightarrow Cyx \right) \right] \rightarrow Dx \right). \tag{19}$$

Indeed, the main quantified phrase of (18) seems to be 'a post', not 'all forum members'. In Quarc, (18) will be formalized as

$$(\forall F, \forall P_x)\, C \rightarrow (x) D. \tag{20}$$

Given our understanding of (18), we would like $\forall P$, and not $\forall F$, to be the governing QA of (20). To accomplish this, we will rely on the notion of a *distributed occurrence*: an occurrence o of a QA is *distributed* in a given formula φ if no subformula of φ contains both o and all its anaphors.[13] The notion of subformula will be given an exact definition in §3. The occurrence of $\forall P$ in (20), for example, is distributed, while that of $\forall F$ is not. We now define the main, or governing, QA of a formula not as the leftmost QA-occurrence in that formula, but as the leftmost *distributed* QA-occurrence. This principle yields the intuitively-correct truth conditions in the case of (20), as well as in many other, similar cases. It also yields the correct truth-conditions in simpler cases, such as that of (16).[14]

Governance is defined here somewhat differently than in (Ben-Yami, 2014). The principle adopted by Ben-Yami makes $\forall F$ the governing QA of (20). Consequently, (18) cannot be formalized as (20) in his logic. I consider this an advantage of the current approach. Other examples seem to fit Ben-Yami's notion of governance better than the one defined here. Technically, Quarc can be developed using either of the two principles of governance (and also, as indicated above, using numerical indices to override the relevant principle of governance).

In PC, the quantifiers' order of priority can be reversed simply by reversing their order of occurrence in the formula. This is what happens when we move from

$$\forall x \exists y Lxy \qquad \text{('Everyone loves someone')} \tag{21}$$

to

$$\exists y \forall x Lxy \qquad \text{('Someone is loved by everyone')}. \tag{22}$$

Since quantifiers in Quarc do not attach to formulas from the outside, a similar effect cannot be obtained in Quarc by simply reversing the order of quantified arguments. As indicated above, the device that will be used for this purpose in Quarc resembles what we find in the English translations of (21) and (22). In English sentences, a change in the quantifiers' order of priority is ordinarily obtained by reversing the order of quantified noun phrases *and* switching between the passive and the active voices (e.g., using 'loved by' instead of 'loves'), or between converse relation expressions (e.g., 'shorter than' instead of 'taller than'). In Quarc, we will associate, with any two-place predicate R, another two-place predicate $R^{2,1}$, that will serve as the converse of R. The predicate $R^{2,1}$ will be called a *reordered form* of R. It will be interpreted, in every interpretation $\mathcal{M}$, as the inverse of the

[13] We use 'all' in 'all its anaphors', as elsewhere in the metalanguage, not as it is used in ordinary discourse (according to the analysis presented above), but as it is standardly used in mathematics.

[14] This is not to say that the principle stated above gives the only possible reading of each quantified English sentence; indeed, in some cases involving ambiguity of scope our principle may not yield the most natural, or plausible, reading.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

relation $R^{\mathcal{M}}$ (where $R^{\mathcal{M}}$ is the denotation in $\mathcal{M}$ of R). In other words: $R^{2,1\mathcal{M}} = \{\langle y, x\rangle : \langle x, y\rangle \in R^{\mathcal{M}}\}$.

To see how reordered forms enable changes in the quantifiers' order of priority, consider again the sentence 'Every man loves some women'. If this sentences is formalized as

$$(\forall M, \exists W)\, L,$$

then 'Some women are loved by every man' will be formalized as

$$(\exists W, \forall M)\, L^{2,1}.$$

It can be verified that the semantic principles already indicated yield the correct truth-conditions when applied to the latter formula.

Many-place predicates in general will be treated similarly. With any n-place predicate R, with $n \geq 2$, we will correlate a set of $n! - 1$ reordered forms: n-place predicates of the form $R^{p_1,\ldots,p_n}$, where p is a nontrivial permutation (i.e., not the identity function) of $\{1, \ldots, n\}$ and $p(i) = p_i$. Reordered forms will be interpreted in any interpretation $\mathcal{M}$ in accordance with the following scheme:

$$R^{p_1,\ldots,p_n\mathcal{M}} = \{\langle x_{p_1}, \ldots, x_{p_n}\rangle : \langle x_1, \ldots, x_n\rangle \in R^{\mathcal{M}}\}.$$

In PC, the negation symbol can either precede a quantifier, or occur between the quantifier and the rest of the formula. We thus have both

$$\neg\exists x\, Hx \qquad \text{('It is not the case that someone is happy')} \tag{23}$$

and

$$\exists x \neg Hx \qquad \text{('Someone isn't happy')}. \tag{24}$$

Since quantifiers in Quarc do not attach to formulas from the outside, the position occupied by the negation symbol in (24) is unavailable. An effect similar to that of (24) can, however, be obtained in Quarc, and in a way similar to what we find in natural languages. The sentence

Some people aren't happy

will be formalized as

$$(\exists P) \neg H. \tag{25}$$

Semantically, a formula

$$(c_1, \ldots, c_n) \neg R,$$

where c_i are individual constants, will be treated as equivalent to

$$\neg (c_1, \ldots, c_n)\, R.$$

This, together with the semantic principles governing quantification in Quarc, will yield the correct truth conditions of formulas such as (25).

As noted in the introduction, Quarc will be capable of straightforward translation of sentences involving defining clauses. The following are two examples of such sentences. In each of them, a noun (the underlined word) is modified by a defining clause (the italicized expression).

John is a <u>man</u> *who knows Mary* (26)

Every <u>man</u> *who knows Mary* knows John (27)

The modified noun, together with the defining clause modifying it, will be represented in Quarc by a *compound predicate*: an expression of the form

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

$$P_x : \psi(x), \tag{28}$$

where P is a one-place predicate and where x is a variable.[15] Intuitively, (28) is to be read as 'P that satisfies the condition $\psi(x)$'. Like primitive one-place predicates, compound predicates will be allowed both in the predicate position and as parts of QAs. Sentences (26) and (27), for instance, will be translated by the following respective formulas:

$$(j)\ \ M_x : [(x, m)\ K] \tag{29}$$

$$(\forall\ M_x : [(x, m)\ K],\ j)\ K. \tag{30}$$

The semantic treatment of formulas such as (29) is straightforward. Intuitively, a formula of the form

$$(c)\ \ P_x : \psi(x)$$

states that 'c is a P that satisfies the condition $\psi(x)$'. Accordingly, that formula will be treated as equivalent to the conjunction

$$(c)P\ \wedge\ \psi(c).$$

Formula (30) involves the quantified argument $\forall\ M_x : [(x, m)\ K]$ ('every man who knows Mary'). To define the truth conditions of formulas involving such QAs, we will first correlate an extension with every compound predicate. This is, again, straightforward: if C is the compound predicate $P_x : \psi(x)$, then the extension $C^{\mathcal{M}}$ of C in $\mathcal{M}$ will be the set of all elements of $P^{\mathcal{M}}$ of which $\psi(x)$ is true in $\mathcal{M}$. Once extensions are assigned in this way, we can extend the semantic principles discussed thus far to formulas such as (30). The latter formula will be true in $\mathcal{M}$ if the extension of

$$M_x : [(x, m)\ K] \qquad (\text{'man who knows Mary'}) \tag{31}$$

in $\mathcal{M}$ is nonempty and if

$$(y, j)\ K \qquad (\text{'}y\text{ knows John'}) \tag{32}$$

is true in $\mathcal{M}$ of every element of that extension. Formula (30) will be false in $\mathcal{M}$ if (32) is false in $\mathcal{M}$ of some element of the extension of (31), and undefined (U) in $\mathcal{M}$ in case that extension is empty.

The following sentence contains a *nested* defining clause:

John is a student who knows every professor whom Mary knows. (33)

This will be translated as

$$(j)\ \ S_x : [\,(x,\ \forall\ P_y : [(m, y)K])\ K]. \tag{34}$$

Note that the semantic principles outlined thus far suffice for the treatment of (34): first, that formula is equivalent to

$$(j)S\ \wedge\ (j,\ \forall\ P_y : [(m, y)K])\ K. \tag{35}$$

[15] In English sentences such as (26) and (27) an argument place in the defining clause is occupied by a word of a special syntactic category: a relative pronoun. It would have been possible to make Quarc resemble English in this respect by introducing a special category of symbols for this role. I believe, however, that there is not much point in doing so. My main reasons are that, first, the employment of relative pronouns in defining clauses is not universal: personal pronouns are sometimes employed in defining clauses in English; and they are often employed in other languages. Second, the introduction of a separate category of symbols to reflect the employment of relative pronouns in defining clauses in English would unnecessarily complicate the syntax of Quarc.

Second, the right conjunct of (35) is true in an interpretation $\mathcal{M}$ iff the extension in $\mathcal{M}$ of

$$P_y:[(m, y)K] \qquad \text{('Professor whom Mary knows')} \tag{36}$$

is nonempty and

$$(j, z)K \qquad \text{('John knows } z\text{')} \tag{37}$$

is true in $\mathcal{M}$ of every element of that extension. The right conjunct of (35) is *false* in $\mathcal{M}$ if (37) is false in $\mathcal{M}$ of some element of the extension in $\mathcal{M}$ of (36). Third, the extension in $\mathcal{M}$ of (36) is the set of all elements of $P^{\mathcal{M}}$ of which $(m, y)K$ is true in $\mathcal{M}$.

As indicated in the introduction, sentences such as (27) ('Every man who knows Mary knows John') cannot be straightforwardly translated into Ben-Yami's (2014) version of Quarc. To translate (27) into Ben-Yami's logic, we need to paraphrase that sentence, probably as

If a man knows Mary, then he knows John, (38)

and only then formalize it, as

$$(\forall M_x, m)\ K \rightarrow (x, j)\ K.\ ^{16} \tag{39}$$

Our translation (30) arguably reflects the syntactic structure of (27) better than Ben-Yami's (39). And assuming the semantic analysis on which Quarc is founded is correct, (30) captures the truth- and falsity-conditions of (27), while Ben-Yami's (39) does not: according to that analysis, (27) cannot be true in case 'man who knows Mary' is empty. But if 'man who knows Marry' is empty while 'man' is not, then (39) is true.

I will close this section with a few comments on anaphors in donkey sentences and on the three-valued framework of Quarc.

Our exact definition of 'formula' will allow anaphors only in cases covered by the principles thus far surveyed. In English, however, there are cases not covered by those principles. As an example, consider the donkey sentence

Every man who digs a hole falls into it. (40)

A straightforward translation of (40) into Quarc would require a formula such as

$$\left(\forall\, M_x:\left[\left(x, \exists H_y\right)D\right],\ y\right)\ F. \tag{41}$$

Expressions such as (41), however, will not be allowed as formulas in Quarc. To translate (40) we will need to paraphrase it first, in one of the following forms, depending on our understanding of the original sentence:

Every man who digs a hole falls into every hole he digs (42)

Every man who digs a hole falls into some hole that he digs. (43)

The last two sentences can be formalized, respectively, as

$$\left(\forall\left[M_x:[(x, \exists H)D]\right]_y,\ \forall H_z:\left[(y, z)\ D\right]\right)F \tag{44}$$

$$\left(\forall\left[M_x:[(x, \exists H)D]\right]_y,\ \exists H_z:\left[(y, z)\ D\right]\right)F. \tag{45}$$

The three-valued framework adopted in this paper results in complications in both the model-theory and the proof-system that will be introduced below. This framework, however, allows Quarc to straightforwardly reflect the basic analysis of quantification on

[16] I am ignoring, for the purpose of the current discussion, certain typographic differences between Ben-Yami's version of Quarc and the current one. Those differences are inconsequential in the present context.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

which it is based. According to that analysis, recall, sentences of the forms 'Some *A*s are *B*s' and 'All *A*s are *B*s' involve the presupposition that the extension of *A* is nonempty. As explained above, this will be reflected in Quarc by letting the translations of such sentences receive the value U ('undefined') whenever *A* has an empty extension.

Ben-Yami (2014) takes a different approach. Since his semantics is bivalent, the above analysis cannot be reflected in his logic in the same way as here. As explained in the introduction, Ben-Yami imposes a global nonemptiness requirement: one-place predicates are simply required to be nonempty, as part of the semantic framework. A consequence of this approach is that $(\exists P)\ P$ ('Some philosophers are philosophers') is—in Ben-Yami's logic—a logical truth. Another consequence is that $\neg\ (\exists W)\ C$ ('It is not the case that some women are witches') logically entails $(\exists C)\ \neg W$ ('Some witches are not women'). Whether or not such consequences are tolerable, the global nonemptiness approach becomes completely inapplicable as soon as we allow (as we do here) defining clauses: since a predicate may be modified by an unsatisfiable defining clause, as in M_x: $x \neq x$ ('a man not identical with himself'), the nonemptiness requirement cannot be imposed. We thus seem to have no real alternative to the three-valued framework—if we want the logic to reflect the analysis of quantification on which we rely.

§3. Syntax. As can be seen from the previous section, Quarc is syntactically and semantically more complex than PC. This will also be evident from the current and next sections. Since Quarc employs devices similar to those of natural languages, which are absent from PC—devices such as negative predication, reordered forms, and defining clauses—the added complexity is to be expected. Additional complexity will result from our employment of a three-valued framework, as will be noticeable in the next two sections. We now proceed to the exact syntactic definitions.

The *logical symbols* are the following:

1. *Identity and nonidentity symbols*: $=, \neq$.
2. *Variables*: $x, y, z, u, v, \ldots, x_1, x_2, \ldots$
3. A *subscript version* of each of the variables: $_x, _y, _z, \ldots$
4. *Superscript indices*: $^1, ^2, ^3, \ldots$
5. *Connectives*: $\neg, \wedge, \vee, \rightarrow$.
6. *Quantifiers*: $\forall, \exists$.
7. *Parentheses*, *square brackets*, *colon*, *comma*, and *superscript comma*: $(,), (,), [,], :, , , '$.

As can be seen, we officially employ two sorts of parentheses, as well as square brackets. Bold parentheses will be used in the construction of truth-functional compounds, square brackets in the construction of compound predicates, and ordinary parentheses for all other purposes. The use of bold parentheses will facilitate a straightforward definition of 'subformula' (Definition 3.7 below). In displaying formulas we will, in general, not distinguish between the three kinds of brackets. Additional notational conventions will be specified at the end of the current section.

The *nonlogical symbols* are:

1. *Individual constants*: $a, b, c, d, \ldots, c_1, c_2, \ldots$
2. For every integer $n \geq 1$, denumerably many *primitive n-place predicates*: $P, Q, R, \ldots, P_1, P_2, \ldots$

A *language* $\mathcal{L}$ is a set whose elements are: (i) all the logical symbols; (ii) denumerably many individual constants; and (iii) some, or none, of the primitive predicates. The

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

elements of a language are its *symbols*. We assume that, for every language $\mathcal{L}$, there are denumerably-many individual constants not in $\mathcal{L}$. The ability to add to every given language denumerably-many new individual constants will be relied on in the completeness proof presented in §6.

The language, $\mathcal{L}$, will be held fixed throughout the rest this section. Unless otherwise stated, by 'symbol', 'individual constant', and 'primitive predicate' we will mean a symbol of $\mathcal{L}$, an individual constant of $\mathcal{L}$ and a predicate of $\mathcal{L}$, respectively. The notions defined below are to be understood as relative to $\mathcal{L}$ despite the omission of explicit reference to the language.

By an *expression* we mean any finite string of symbols. Expressions of length 1 will be identified with their single component. To emphasize the fact that an expression φ includes as a substring an expression E we may display φ as $\varphi(E)$. If o is an occurrence of some expression within an expression φ, and E is any expression, then by $\varphi[o/E]$ we will mean the expression resulting from φ on replacing o with E. In case several expression-occurrences $o_1, \ldots, o_n$ are replaced by $E_1, \ldots, E_n$, respectively, we will write $\varphi[o_1/E_1, \ldots, o_n/E_n]$. Sometimes we would like to replace all the occurrences of an expression E_1 in φ with another expression E_2. We refer to the expression resulting in this way as $\varphi[E_1/E_2]$. The same notation ($\varphi[E_1/E_2]$) will also have the following use: it will denote the expression resulting from $\varphi(E_1)$ on replacing one particular occurrence of E_1 with E_2, where the relevant occurrence of E_1 is clear from the context.

DEFINITION 3.1 (Reordered form). *If R is an n-place primitive predicate, $n \geq 2$, p is a nontrivial permutation (i.e., not the identity permutation) of $\{1, \ldots, n\}$, and $p(i) = p_i$ for all i, then the expression $R^{p_1,\ldots,p_n}$ is a* reordered form *of R.*

Thus, if R is a 3-place primitive predicate, its reordered forms are $R^{1,3,2}$, $R^{2,1,3}$, $R^{2,3,1}$, $R^{3,1,2}$, and $R^{3,2,1}$.

DEFINITION 3.2 (Basic formula). *An expression φ is a* basic formula *if it has one of the following forms, where c_i are individual constants, and R is a k-place primitive predicate ($k \geq 1$) or a reordered form of such*:

1. $c_1 = c_2$
2. $c_1 \neq c_2$
3. $(c_1, \ldots, c_k)\, R$
4. $(c_1, \ldots, c_k)\, \neg R$

Basic formulas are formulas. We will define the set of *all* formulas as the closure of the set of basic formulas under certain syntactic operations that will be introduced below. The following seven definitions will be relied on in the definition of those operations.

We start with the definition of *compound predicate*. Officially, compound predicates will have the form $P_x : [\psi(x)]$, where $\psi(x)$ results from a formula $\psi(c)$ on replacing c with x. The square brackets will be relied on in the definitions below, but will usually be omitted when displaying formulas. Since the notion of formula is not yet available at this stage, we will first define a more general notion of compound predicate, where $\psi(c)$ is an expression containing c, but not necessarily a formula. The notions of predicate and of quantified argument will then be treated in the same way.

DEFINITION 3.3 (Compound ψ-predicate). *If $\psi(c)$ is an expression containing an individual constant c, P is a one-place primitive predicate, x is a variable not occurring in $\psi(c)$, $\psi(x)$ is $\psi[c/x]$, and in the expression $[\psi(x)]$ the displayed square brackets constitute a*

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

matching pair,[17] *then the expression* $P_x : [\psi(x)]$ *is a* compound ψ-predicate*, and* $\psi(x)$ *is its* defining clause*.*

Example. If $\psi(c)$ is $(c_y, y)L$ ('Charlie loves himself'), then $M_x : [(x_y, y)L]$ ('man who loves himself') is a compound ψ-predicate.

DEFINITION 3.4 (ψ-predicate). *An expression* Q *is a* k-place ψ-predicate *if either* (*i*) Q *is a primitive* k*-place predicate; or* (*ii*) $k = 1$ *and* Q *is a compound* ψ*-predicate; or* (*iii*) $k \geq 2$ *and* Q *is a reordered form of a primitive* k*-place predicate.*

DEFINITION 3.5 (Quantified ψ-argument; ψ-argument). *If* P *is a one-place* ψ*-predicate* (*either primitive or compound*) *then the expressions* $\forall P$ *and* $\exists P$ *are* quantified ψ-arguments*. An expression* α *is a* ψ-argument *if it is either a quantified* ψ*-argument or an individual constant.*

DEFINITION 3.6 (Anaphor). *An occurrence* o_x *of a variable* x *in an expression* φ *is* anaphoric on *an occurrence* o_α *of a* ψ*-argument* α *in* φ *if the following conditions hold*:

1. o_α *is to the left of* o_x.
2. *The subscript variable* $_x$ *immediately succeeds* o_α.
3. *The subscript variable* $_x$ *does not immediately succeed any* ξ*-argument* (*for any expression* ξ) *between* o_α *and* o_x.

In this case we may also say that o_x *is an* anaphor of o_α*, and that* o_α *is the* source of o_x*, in* φ*. In cases where no confusion would arise, we may also say that* o_x *is an anaphor of* α *in* φ.

DEFINITION 3.7 (Subformula). *If* φ *is any expression, then by a* subformula *of* φ *we will mean any expression* ψ *such that the expression* (ψ) *is part of* φ*, and the displayed bold parentheses constitute a matching pair.*

DEFINITION 3.8 (Distributed occurrence). *An occurrence* o_α *of a* ψ*-argument* α *in an expression* φ *is* distributed *in* φ *if the following conditions hold*:

1. o_α *does not lie within any defining clause.*
2. *No subformula of* φ *contains both* o_α *and all its anaphors.*

DEFINITION 3.9 (Governance). *An occurrence* o_κ *of a quantified* ψ*-argument* κ *in an expression* φ governs φ *if* o_κ *is the leftmost distributed occurrence of such an argument in* φ (*in other words: if* o_κ *is distributed in* φ *and it is not the case that for some expression* ξ*, there is in* φ *a distributed occurrence of a quantified* ξ*-argument to the left of* o_κ).

The following four syntactic operations, O1–O4, will be relied on in the definition of formula:

[17] The relation 'p and q constitute a matching pair', on occurrences of square brackets in a given expression, can be defined by induction on the number n of additional occurrences of square brackets between p and q: (i) if $n = 0$, then p and q constitute a matching pair iff (*) the leftmost of p, q is an occurrence of the symbol [and the rightmost of p, q is an occurrence of the symbol]. (ii) If $n > 0$, then, assuming the relation to be defined for every $k < n$, p and q constitute a matching pair iff the condition (*) is met and for every square-bracket occurrence p' between p and q there is another square-bracket occurrence q' between p and q such that p' and q' constitute a matching pair.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

O1 (Compound predicates)**.** *From an expression* $\psi(c_1)$ *obtain the expressions* $(c_2)\ P_x : [\psi(x)]$ *and* $(c_2)\ \neg P_x : [\psi(x)]$*, where:*

1. c_1*,* c_2 *are individual constants.*
2. $\psi(x)$ *is* $\psi[c_1/x]$*.*
3. $P_x : [\psi(x)]$ *is a compound* ψ*-predicate.*

Example. The expression (in fact: formula) $(d)\ M_x : [(x, m)\ L]$ ('David is a man who loves Mary') is obtained from $(j, m)L$ ('John loves Mary') by the operation O1.

O2 (Truth-functional compounds)**.** *From a pair* φ*,* ψ *of expressions, none of which contains anaphors of individual constants, obtain the expressions*: $\neg(\varphi)$, $(\varphi)\wedge(\psi)$, $(\varphi)\vee(\psi)$, $(\varphi) \rightarrow (\psi)$.

O3 (Constant anaphors)**.** *From an expression* $\psi(c)$ *obtain the expression* $\psi[o_1/c_x, o_2/x, \ldots, o_m/x]$*, where*:

1. c *is an individual constant*; x *is a variable not occurring in* ψ*.*
2. $m \geq 2$*.*
3. $o_1, o_2, \ldots, o_m$ *are occurrences of* c *in* ψ*, ordered from left to right.*
4. *None of* $o_1, o_2, \ldots, o_m$ *has anaphors in* ψ*.*

Example. The expression $(j_x, x, m)\ \neg I$ ('John did not introduce himself to Mary') is obtained from $(j,\ j, m)\ \neg I$ ('John did not introduce John to Mary') by O3.

O4 (Quantified arguments)**.** *From expressions* $\varphi(c)$ *and* ψ *obtain the expression* $\varphi(q\,P)$*, where*:

1. c *is an individual constant occurring in* $\varphi(c)$ *exactly once.*
2. q *is either* $\forall$ *or* $\exists$*.*
3. $q\,P$ *is a quantified* ψ*-argument* (P *is either a primitive one-place predicate or a compound* ψ*-predicate of the form* $F_x : [\psi(x)]$)*.*
4. *The expression* $\varphi(q\,P)$ *results from* $\varphi(c)$ *on replacing* c *with* $q\,P$*.*
5. *The occurrence of* $q\,P$ *replacing* c *governs the expression* $\varphi(q\,P)$*.*
6. *No individual constant has anaphors in* $\varphi(q\,P)$*.*
7. *No variable occurs both in* $\varphi(c)$ *and in* $q\,P$*.*

Example. If M is a primitive one-place predicate, then the expression $\forall M$ is a quantified ψ-predicate for every expression ψ. Hence, for every expression ψ, the expression $((\forall M_x, \forall W_y)L) \rightarrow ((y, x)L^{2,1})$ ('If a man loves a women, then she is loved by him') results from $((j_x, \forall W_y)L) \rightarrow ((y, x)L^{2,1})$ ('If John loves a women, then she is loved by him') and ψ by O4.

DEFINITION 3.10 (Formula). *The set of* formulas *is the closure under O1–O4 of the set of basic formulas.*

Now that the notion of formula has been defined, we can easily define restricted notions of compound predicate, predicate, quantified argument, and argument, removing the relativization to an expression ψ:

DEFINITION 3.11 (Compound predicate; predicate; quantified argument; argument). *An expression* E *is a* compound predicate (predicate, quantified argument, argument) *if there is a formula* ψ *such that* E *is a compound* ψ*-predicate* (ψ*-predicate, quantified* ψ*-argument,* ψ*-argument*).

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

In displaying formulas, it will be convenient to adopt the following conventions:

1. By $\varphi(qP)$ we will mean a formula φ generated by the operation O4 and governed by an occurrence o_{qP} of qP. Also, by $\varphi[qP/\beta]$, and sometimes just $\varphi(\beta)$, we will mean $\varphi[o_{qP}/\beta]$.
2. We may use parentheses of different kinds, as well as add parentheses, to increase readability.
3. We will not write parentheses in bold font, and will omit parentheses where this is unlikely to cause confusion. We will also usually omit the square brackets around the defining clauses of compound predicates.

§4. Semantics. As noted in §2, interpretations in Quarc will not be equipped with a domain of quantification; rather, they will simply be interpretation *functions*. The need for domains of quantification will not arise here, since the pluralities needed for quantification will always be introduced by the predicates to which quantifiers attach. When reference is made to the domain of an interpretation $\mathcal{M}$, what will be meant is the domain of $\mathcal{M}$ *as a function*. The language $\mathcal{L}$ remains fixed throughout this section.

The following definition relies on the notions of function, domain of a function, and k-tuple. Note that these three notions are set-theoretically definable without prior specification of a set, or domain, from which the values of a function, or the components of a k-tuple, are to be taken.[18]

DEFINITION 4.1 (Interpretation). *An* interpretation *for $\mathcal{L}$ is a function $\mathcal{M}$ for which the following conditions hold* (*where $E^{\mathcal{M}}$ is the image under $\mathcal{M}$ of E*):

1. *The domain of $\mathcal{M}$* (*as a function*) *is the set of all individual constants, primitive predicates, and reordered forms of predicates in $\mathcal{L}$.*
2. *For every k-place predicate R in $\mathcal{L}$, $R^{\mathcal{M}}$ is a set of k-tuples.*
3. *For every reordered form $R^{p_1,\ldots,p_k}$ in $\mathcal{L}$ ($k \geq 2$), $R^{p_1,\ldots,p_k\mathcal{M}} = \{\langle x_{p_1}, \ldots, x_{p_k}\rangle : \langle x_1, \ldots, x_k\rangle \in R^{\mathcal{M}}\}$.*[19]

DEFINITION 4.2 (Truth-value in an interpretation). *We now inductively define the truth-value $\varphi^{\mathcal{M}}$ of a formula φ in an interpretation $\mathcal{M}$. The truth-value $\varphi^{\mathcal{M}}$ will always be either* T (*true*)*,* F (*false*)*, or* U (*undefined*)*.*

DEFINITION 4.2.1 (Basic formulas).

$$[c_1 = c_2]^{\mathcal{M}} = \begin{cases} \mathrm{T} & \text{if } {c_1}^{\mathcal{M}} = {c_2}^{\mathcal{M}}, \\ \mathrm{F} & \text{otherwise.} \end{cases}$$

$$[c_1 \neq c_2]^{\mathcal{M}} = \begin{cases} \mathrm{T} & \text{if } {c_1}^{\mathcal{M}} \neq {c_2}^{\mathcal{M}}, \\ \mathrm{F} & \text{otherwise.} \end{cases}$$

[18] A *function* is a set f of ordered pairs such that, for every x, y, and z, if $\langle x, y\rangle, \langle x, z\rangle \in f$, then $y = z$. The *domain* of a function f is the set $\{x :$ for some y, $\langle x, y\rangle \in f\}$. A *$k$-tuple* is a function whose domain is the set $\{1, \ldots, k\}$.

[19] In ordinary model-theory, the denotations of individual constants are required to belong to the domain of quantification of the relevant interpretation. Nothing of this sort needs to (or can) be required here, as Quarc-interpretations lack domains of quantification.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

$$[(c_1, \ldots, c_k)R]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } \langle c_1{}^{\mathcal{M}}, \ldots, c_k{}^{\mathcal{M}} \rangle \in R^{\mathcal{M}}, \\ \text{F} & \text{otherwise}. \end{cases}$$

$$[(c_1, \ldots, c_k)\neg R]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } \langle c_1{}^{\mathcal{M}}, \ldots, c_k{}^{\mathcal{M}} \rangle \notin R^{\mathcal{M}}, \\ \text{F} & \text{otherwise}. \end{cases}$$

c_i here are any individual constants, $k \geq 1$, and R is any primitive k-place predicate or a reordered form of such.

DEFINITION 4.2.2 (Compound predicates). $[(c)P_x : \psi(x)]^{\mathcal{M}} = [(c)P \wedge \psi(c)]^{\mathcal{M}}$, *where, for some individual constant d, the expression $\psi(x)$ is $\psi[d/x]$ and $\psi(c)$ is $\psi[d/c]$. $[(c)\neg P_x : \psi(x)]^{\mathcal{M}}$ is the opposite of the truth value in $\mathcal{M}$ of $(c)P \wedge \psi(c)$: F if the latter is* T, T *if it is* F, *and* U *otherwise.*[20]

DEFINITION 4.2.3 (Truth-functional compounds). *The truth-values of these in every interpretation are determined in accordance with Kleene's strong truth-tables*:

$$[\neg\varphi]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } \varphi^{\mathcal{M}} = \text{F}, \\ \text{F} & \text{if } \varphi^{\mathcal{M}} = \text{T}, \\ \text{U} & \text{otherwise}. \end{cases}$$

$$[\varphi \wedge \psi]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } \varphi^{\mathcal{M}} = \psi^{\mathcal{M}} = \text{T}, \\ \text{F} & \text{if at least one of } \varphi^{\mathcal{M}}, \psi^{\mathcal{M}} \text{ is F}, \\ \text{U} & \text{otherwise}. \end{cases}$$

$$[\varphi \vee \psi]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if at least one of } \varphi^{\mathcal{M}}, \psi^{\mathcal{M}} \text{ is T}, \\ \text{F} & \text{if } \varphi^{\mathcal{M}} = \psi^{\mathcal{M}} = \text{F}, \\ \text{U} & \text{otherwise}. \end{cases}$$

$$[\varphi \rightarrow \psi]^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } \varphi^{\mathcal{M}} = \text{F or } \psi^{\mathcal{M}} = \text{T}, \\ \text{F} & \text{if } \varphi^{\mathcal{M}} = \text{T and } \psi^{\mathcal{M}} = \text{F}, \\ \text{U} & \text{otherwise}. \end{cases}$$

DEFINITION 4.2.4 (Constant anaphors). *If a formula φ results from a formula ψ by the operation O3 of §3, then $\varphi^{\mathcal{M}} = \psi^{\mathcal{M}}$.*

To define the truth conditions of quantified formulas we will rely on the following three definitions:

DEFINITION 4.2.5 ($\frac{\alpha}{c}$-variant). *For any object α, any interpretation $\mathcal{M}$, and any individual constant c, the* $\frac{\alpha}{c}$-variant *of $\mathcal{M}$, or: $\mathcal{M}^{\alpha}_{c}$, is the interpretation that sends c to α and that coincides with $\mathcal{M}$ on any other input.*

DEFINITION 4.2.6 (True of, false of, undefined for). *If $\psi(c)$ is a formula not containing x and $\psi(x)$ is the expression $\psi[c/x]$, then: $\psi(x)$* is *true of* (*false of, undefined for*) *α* in *$\mathcal{M}$* if *$\psi(c)$* is true (false, undefined) in *$\mathcal{M}^{\alpha}_{c}$*.[21]

[20] It is straightforward to verify that the definition is independent of the choice of d.

[21] It is straightforward to verify that the definition, for a given $\psi(x)$, is independent of the choice of c.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

DEFINITION 4.2.7 (Extensions of compound predicates). *The extension in $\mathcal{M}$ of a compound predicate $P_x : \psi(x)$ is the set $\{\alpha \in P^{\mathcal{M}} : \psi(x)$ is true of α in $\mathcal{M}\}$. The extension in $\mathcal{M}$ of a compound predicate C will be referred to as $C^{\mathcal{M}}$.*

Example. If $B^{\mathcal{M}}$ is the set of all integers and $L^{\mathcal{M}}$ is the set $\{\langle x, y\rangle \in \mathbb{R}^2 : x + y = xy\}$, then the extension of $B_x : (x_y, y)L$ in $\mathcal{M}$ is the set of all integers n such that $n + n = nn$ (i.e., the set $\{0, 2\}$).

DEFINITION 4.2.8 (Quantifiers). *If P is any one-place predicate, q is a quantifier, $\varphi(q\,P)$ does not contain x and results from $\varphi(c)$ and a formula ψ by the operation O4 of §3, and $\varphi(x)$ is $\varphi[c/x]$, then*:

1. *In case q is* $\forall$:

$$\varphi(\forall P)^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } P^{\mathcal{M}} \neq \emptyset \text{ and } \varphi(x) \text{ is true in } \mathcal{M} \text{ of every element of } P^{\mathcal{M}}, \\ \text{F} & \text{if } P^{\mathcal{M}} \neq \emptyset \text{ and } \varphi(x) \text{ is false in } \mathcal{M} \text{ of some element of } P^{\mathcal{M}}, \\ \text{U} & \text{otherwise}. \end{cases}$$

2. *In case q is* $\exists$:

$$\varphi(\exists P)^{\mathcal{M}} = \begin{cases} \text{T} & \text{if } P^{\mathcal{M}} \neq \emptyset \text{ and } \varphi(x) \text{ is true in } \mathcal{M} \text{ of some element of } P^{\mathcal{M}}, \\ \text{F} & \text{if } P^{\mathcal{M}} \neq \emptyset \text{ and } \varphi(x) \text{ is false in } \mathcal{M} \text{ of every element of } P^{\mathcal{M}}, \\ \text{U} & \text{otherwise}. \end{cases}$$

DEFINITION 4.3 (Models). *An interpretation $\mathcal{M}$ is a* model of *a formula φ if φ is true in $\mathcal{M}$. $\mathcal{M}$ is a* model of *a set Γ of formulas if it is a model of each $\varphi \in \Gamma$.*

DEFINITION 4.4 (Entailment). *A set Γ of formulas* entails *a formula φ* in *$\mathcal{L}$ (in symbols: $\Gamma \vDash_{\mathcal{L}} \varphi$) if every model of Γ is a model of φ.*

Example. $(\forall S)\ P$ entails $(\exists S)\ P$.

Proof. If $\mathcal{M}$ is a model of $(\forall S)\ P$ then $S^{\mathcal{M}}$ is nonempty and $(x)P$ is true of all its elements. It follows that $(x)P$ is true of at least one element of $S^{\mathcal{M}}$, and thus—that $(\exists S)\ P$ is true in $\mathcal{M}$. □

The inference from $(\forall S)\ P$ to $(\exists S)\ P$ is part of the Aristotelian Square of Opposition (when 'Every S is P' is translated as $(\forall S)\ P$ and 'Some S is P' – as $(\exists S)\ P$). It is straightforward to verify that *all* the inferences of the Square are similarly validated in Quarc.[22]

§5. Proof system. The proof system presented below operates on sequents, but is more similar to a natural deduction system in its inference rules than to a sequent calculus. The

[22] Unlike the inferences of the Square of Opposition, not all conversion principles of Aristotelian logic are preserved in Quarc. Consider, for instance, the inference from 'No S is P' (or, in Quarc: $(\forall S)\ \neg P$) to 'No P is S' (in Quarc: $(\forall P)\ \neg S$). Any interpretation $\mathcal{M}$ with $S^{\mathcal{M}} \neq \emptyset$ and $P^{\mathcal{M}} = \emptyset$ is a model of this inference's premise, but not of its conclusion.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

system is considerably more complex than the ones usually given for PC. The complexity results in large part from the three-valued framework adopted here. In that framework, the classical ¬-introduction rule cannot be employed; instead, we use separate introduction and elimination rules for various combinations of ¬ with other symbols. Thus, for instance, in the case of conjunction, we will employ not a single introduction rule and a single elimination rule, but introduction and elimination rules for each of $\varphi \wedge \psi$ and $\neg(\varphi \wedge \psi)$.[23]

The language, $\mathcal{L}$, is held fixed throughout this section.

Our proof system will operate on sequents of the form $\Gamma \Rightarrow \varphi$, where Γ is a (possibly empty) set of formulas, and φ is a single formula. The elements of Γ (if there are any) are the *antecedent formulas* of the sequent; φ is the sequent's *succedent*. The antecedent formulas may be written in any of the usual forms, e.g., $\Gamma, \psi \Rightarrow \varphi$, or $\psi_1, \psi_2, \psi_3 \Rightarrow \varphi$.

5.1. Inferences; deductions. An *inference* in $\mathcal{L}$ is an array of sequents of the form

$$\frac{\Gamma_1 \Rightarrow \varphi_1 \ldots \Gamma_n \Rightarrow \varphi_n}{\Gamma \Rightarrow \varphi}$$

conforming to one of the inference rules specified below. The sequents $\Gamma_i \Rightarrow \varphi_i$ are called the inference's *upper sequents*; $\Gamma \Rightarrow \varphi$ is called its *lower sequent*.

A *deduction* in $\mathcal{L}$ is an array of inferences arranged in the usual tree form. Each deduction starts with finitely many *initial sequents* (see below) and ends with a single *end sequent*. A sequent $\Gamma \Rightarrow \varphi$ is *deducible* in $\mathcal{L}$ if there is a deduction in $\mathcal{L}$ of which that sequent is the end sequent. Such a deduction will be called a *deduction in $\mathcal{L}$ of* $\Gamma \Rightarrow \varphi$. Instead of saying that $\Gamma \Rightarrow \varphi$ is deducible in $\mathcal{L}$, we may say that φ is *deducible from* Γ in $\mathcal{L}$, and write $\Gamma \vdash_{\mathcal{L}} \varphi$.

5.1.1. Initial sequents.

1. $\Gamma \Rightarrow \varphi$, where $\varphi \in \Gamma$;
2. $\Gamma \Rightarrow c = c$, where c is any indidivual constant;
3. $\Gamma \Rightarrow \varphi \vee \neg\varphi$, where φ is a basic formula.

5.1.2. Inference rules. Throughout this subsection, unless otherwise specified, c, c_i, and d are any individual constants; Q is any one-place predicate (either primitive or compound); P is any one-place primitive predicate; R is any n-place predicate, $n \geq 2$; φ, ψ, and ξ are formulas; and Γ, Γ_i are sets of formulas.

Identity:

$$\text{(SV)}\quad \frac{\Gamma_1 \Rightarrow \varphi(c) \qquad \Gamma_2 \Rightarrow c = d}{\Gamma_1, \Gamma_2 \Rightarrow \varphi(d)},$$

where $\varphi(d)$ results from $\varphi(c)$ on replacing some, or all, of the occurrences of c with d.

Reordered forms:

$$\text{(RF1)}\quad \frac{\Gamma \Rightarrow (c_{p_1}, \ldots, c_{p_n})R^{p_1,\ldots,p_n}}{\Gamma \Rightarrow (c_{r_1}, \ldots, c_{r_n})R^{r_1,\ldots,r_n}} \qquad \text{(RF2)}\quad \frac{\Gamma \Rightarrow \neg(c_{p_1}, \ldots, c_{p_n})R^{p_1,\ldots,p_n}}{\Gamma \Rightarrow \neg(c_{r_1}, \ldots, c_{r_n})R^{r_1,\ldots,r_n}},$$

where p and r are any two permutations of $\{1, \ldots, n\}$, and where $R^{1,\ldots,n}$ is just R.

[23] A similar thing happens with proof systems for Kleene's first-order logic. See (Kearns, 1979).

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

Negative predication:

$$\text{(NPI1)}\ \frac{\Gamma \Rightarrow \neg(c = d)}{\Gamma \Rightarrow c \neq d} \qquad \text{(NPE1)}\ \frac{\Gamma \Rightarrow c \neq d}{\Gamma \Rightarrow \neg(c = d)}$$

$$\text{(NPI2)}\ \frac{\Gamma \Rightarrow \neg(c_1, \ldots, c_n)R}{\Gamma \Rightarrow (c_1, \ldots, c_n)\neg R} \qquad \text{(NPE2)}\ \frac{\Gamma \Rightarrow (c_1, \ldots, c_n)\neg R}{\Gamma \Rightarrow \neg(c_1, \ldots, c_n)R}$$

$$\text{(NPI3)}\ \frac{\Gamma \Rightarrow (c_1, \ldots, c_n)R}{\Gamma \Rightarrow \neg(c_1, \ldots, c_n)\neg R} \qquad \text{(NPE3)}\ \frac{\Gamma \Rightarrow \neg(c_1, \ldots, c_n)\neg R}{\Gamma \Rightarrow (c_1, \ldots, c_n)R},$$

where $n \geq 1$ and R is any n-place predicate (possibly compound, in case $n = 1$, and possibly a reordered form, in case $n > 1$).[24]

Compound predicates:

$$\text{(CPI1)}\ \frac{\Gamma \Rightarrow (c)P \wedge \psi(c)}{\Gamma \Rightarrow (c)P_x{:}\psi(x)} \qquad \text{(CPE1)}\ \frac{\Gamma \Rightarrow (c)P_x{:}\psi(x)}{\Gamma \Rightarrow (c)P \wedge \psi(c)}$$

$$\text{(CPI2)}\ \frac{\Gamma \Rightarrow \neg[(c)P \wedge \psi(c)]}{\Gamma \Rightarrow \neg[(c)P_x{:}\psi(x)]} \qquad \text{(CPE2)}\ \frac{\Gamma \Rightarrow \neg[(c)P_x{:}\psi(x)]}{\Gamma \Rightarrow \neg[(c)P \wedge \psi(c)]},$$

where $\psi(c)$ and $\psi(x)$ result from a formula $\psi(d)$ on substituting c and x (respectively) for every occurrence of d.

Anaphors:

$$\text{(AI)}\ \frac{\Gamma \Rightarrow \psi(c)}{\Gamma \Rightarrow \psi[o_1/c_x, o_2/x, \ldots, o_m/x]} \qquad \text{(AE)}\ \frac{\Gamma \Rightarrow \psi[o_1/c_x, o_2/x, \ldots, o_m/x]}{\Gamma \Rightarrow \psi(c)}$$

where $\psi[o_1/c_x, o_2/x, \ldots, o_m/x]$ results from $\psi(c)$ by the operation O3 of §3.

Conjunction:

$$\text{(}\wedge\text{I1)}\ \frac{\Gamma_1 \Rightarrow \varphi \qquad \Gamma_2 \Rightarrow \psi}{\Gamma_1, \Gamma_2 \Rightarrow \varphi \wedge \psi} \qquad \text{(}\wedge\text{E1)}\ \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \varphi} \qquad \text{(}\wedge\text{E2)}\ \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \psi}$$

$$\text{(}\wedge\text{I2)}\ \frac{\Gamma \Rightarrow \neg\varphi}{\Gamma \Rightarrow \neg(\varphi \wedge \psi)} \qquad \text{(}\wedge\text{I3)}\ \frac{\Gamma \Rightarrow \neg\psi}{\Gamma \Rightarrow \neg(\varphi \wedge \psi)} \qquad \text{(}\wedge\text{E3)}\ \frac{\Gamma \Rightarrow \neg(\varphi \wedge \psi)}{\Gamma \Rightarrow \neg\varphi \vee \neg\psi}$$

Implication:

$$\text{(}\rightarrow\text{I1)}\ \frac{\Gamma \Rightarrow \neg\varphi \vee \psi}{\Gamma \Rightarrow \varphi \rightarrow \psi} \qquad \text{(}\rightarrow\text{E1)}\ \frac{\Gamma \Rightarrow \varphi \rightarrow \psi}{\Gamma \Rightarrow \neg\varphi \vee \psi} \qquad \text{(}\rightarrow\text{I2)}\ \frac{\Gamma \Rightarrow \varphi \wedge \neg\psi}{\Gamma \Rightarrow \neg(\varphi \rightarrow \psi)} \qquad \text{(}\rightarrow\text{E2)}\ \frac{\Gamma \Rightarrow \neg(\varphi \rightarrow \psi)}{\Gamma \Rightarrow \varphi \wedge \neg\psi}$$

[24] Given the rest of the inference rules, (NPI3) and (NPE3) are redundant whenever R is a primitive predicate: their lower sequents can be derived from their upper sequents using the relevant initial sequents involving R. This, however, is not so in case R is a compound (one-place) predicate.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

Disjunction:

$$(\vee\text{I1})\ \frac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \varphi \vee \psi} \qquad (\vee\text{E1})\ \frac{\Gamma \Rightarrow \neg(\varphi \vee \psi)}{\Gamma \Rightarrow \neg\varphi} \qquad (\vee\text{I2})\ \frac{\Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \vee \psi} \qquad (\vee\text{E2})\ \frac{\Gamma \Rightarrow \neg(\varphi \vee \psi)}{\Gamma \Rightarrow \neg\psi}$$

$$(\vee\text{I3})\ \frac{\Gamma_1 \Rightarrow \neg\varphi \qquad \Gamma_2 \Rightarrow \neg\psi}{\Gamma_1, \Gamma_2 \Rightarrow \neg(\varphi \vee \psi)} \qquad (\vee\text{E3})\ \frac{\Gamma_1, \varphi \Rightarrow \xi \qquad \Gamma_2, \psi \Rightarrow \xi \qquad \Gamma_3 \Rightarrow \varphi \vee \psi}{\Gamma_1, \Gamma_2, \Gamma_3 \Rightarrow \xi}$$

$$(\text{DS})\ \frac{\Gamma_1 \Rightarrow \neg\varphi \vee \psi \qquad \Gamma_2 \Rightarrow \varphi}{\Gamma_1, \Gamma_2 \Rightarrow \psi}$$

Double negation:

$$(\neg\text{I})\ \frac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \neg\neg\varphi} \qquad (\neg\text{E})\ \frac{\Gamma \Rightarrow \neg\neg\varphi}{\Gamma \Rightarrow \varphi}$$

Quantifiers:

In the following quantifier rules, $\varphi(\forall Q)$ is a formula generated by O4 and governed by an occurrence of $\forall Q$, and $\varphi(c)$ results from that formula on replacing the governing occurrence of $\forall Q$ with c. Similarly for $\varphi(\exists Q)$ and $\varphi(c)$.

The next two rules express the following idea: whatever can be inferred from the assumption that an arbitrary, particular, Q is such and such can be inferred from the assumption that some Qs are such and such, or that all Qs are such and such. This principle, in turn, reflects the basic analysis of quantification on which Quarc is founded (see §2). The need for two such rules, rather than just one, is due to the three-valued framework.

$$(q\text{E1})\ \frac{\Gamma_1, (c)Q, \varphi(c) \Rightarrow \psi \qquad \Gamma_2 \Rightarrow \varphi(qQ)}{\Gamma_1, \Gamma_2 \Rightarrow \psi} \qquad (q\text{E2})\ \frac{\Gamma_1, (c)Q, \neg\varphi(c) \Rightarrow \psi \qquad \Gamma_2 \Rightarrow \neg\varphi(qQ)}{\Gamma_1, \Gamma_2 \Rightarrow \psi},$$

where q is a either $\forall$ or $\exists$, and c does not occur in $\varphi(qQ)$, in ψ, or in any element of Γ_1.

$\forall$:

$$(\forall\text{I1})\ \frac{\Gamma_1, (c)Q \Rightarrow \psi \vee \varphi(c) \qquad \Gamma_2 \Rightarrow (\exists Q)Q}{\Gamma_1, \Gamma_2 \Rightarrow \psi \vee \varphi(\forall Q)} \qquad (\forall\text{E})\ \frac{\Gamma_1 \Rightarrow \varphi(\forall Q) \qquad \Gamma_2 \Rightarrow (c)Q}{\Gamma_1, \Gamma_2 \Rightarrow \varphi(c)}$$

$$(\forall\text{I2})\ \frac{\Gamma_1 \Rightarrow (c)Q \qquad \Gamma_2 \Rightarrow \neg\varphi(c)}{\Gamma_1, \Gamma_2 \Rightarrow \neg\varphi(\forall Q)},$$

where, in ($\forall$I1), c does not occur in φ, in ψ, or any element of Γ_1.

$\exists$:

$$(\exists\text{I1})\ \frac{\Gamma_1 \Rightarrow (c)Q \qquad \Gamma_2 \Rightarrow \varphi(c)}{\Gamma_1, \Gamma_2 \Rightarrow \varphi(\exists Q)} \qquad (\exists\text{E})\ \frac{\Gamma_1 \Rightarrow \neg\varphi(\exists Q) \qquad \Gamma_2 \Rightarrow (c)Q}{\Gamma_1, \Gamma_2 \Rightarrow \neg\varphi(c)}$$

$$(\exists\text{I2})\ \frac{\Gamma_1, (c)Q \Rightarrow \psi \vee \neg\varphi(c) \qquad \Gamma_2 \Rightarrow (\exists Q)Q}{\Gamma_1, \Gamma_2 \Rightarrow \psi \vee \neg\varphi(\exists Q)},$$

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

where, in (∃I2), c does not occur in φ, in ψ, or any element of Γ_1.

Structural rules:

$$\text{(CUT)}\quad \frac{\Gamma_1 \Rightarrow \varphi \qquad \Gamma_2, \varphi \Rightarrow \psi}{\Gamma_1, \Gamma_2 \Rightarrow \psi} \qquad \text{(THIN)}\quad \frac{\Gamma_1 \Rightarrow \varphi}{\Gamma_1, \Gamma_2 \Rightarrow \varphi}$$

We close this subsection with a simple example. Consider the sequent:

$$(\forall M_x, x)L,\ (j)M \ \Rightarrow\ (j_x, x)L,$$

representing the argument

Every man loves himself; John is a man; hence, John loves himself.

This sequent can be deduced by a single application of the rule (∀E):

$$\frac{(\forall M_x, x)L \ \Rightarrow\ (\forall M_x, x)L \qquad (j)M \ \Rightarrow\ (j)M}{(\forall M_x, x)L,\ (j)M \ \Rightarrow\ (j_x, x)L}$$

The upper sequents here are, of course, initial sequents.

5.2. Derived rules. A *derived inference rule* is a scheme of the form

$$\frac{\Gamma_1 \Rightarrow \varphi_1 \ldots \Gamma_n \Rightarrow \varphi_n}{\Gamma \Rightarrow \varphi}$$

such that, whenever the sequents $\Gamma_i \Rightarrow \varphi_i$ are deducible in $\mathcal{L}$, then so is $\Gamma \Rightarrow \varphi$.

The following are derived inference rules:

$$\text{(SYM)}\quad \frac{\Gamma \Rightarrow c = d}{\Gamma \Rightarrow d = c} \qquad \text{(TRA)}\quad \frac{\Gamma_1 \Rightarrow c = d \qquad \Gamma_2 \Rightarrow d = e}{\Gamma_1, \Gamma_2 \Rightarrow c = e} \qquad \text{(MP)}\quad \frac{\Gamma_1 \Rightarrow \varphi \qquad \Gamma_2 \Rightarrow \varphi \to \psi}{\Gamma_1, \Gamma_2 \Rightarrow \psi}$$

$$\text{(EFQ)}\quad \frac{\Gamma_1 \Rightarrow \varphi \qquad \Gamma_2 \Rightarrow \neg\varphi}{\Gamma_1, \Gamma_2 \Rightarrow \psi} \qquad \text{(DIS)}\quad \frac{\Gamma \Rightarrow \varphi \vee \varphi}{\Gamma \Rightarrow \varphi} \qquad (\forall/\exists)\quad \frac{\Gamma \Rightarrow \varphi(\forall Q)}{\Gamma \Rightarrow \varphi[\forall Q/\exists Q]} \qquad (\exists/\forall)\quad \frac{\Gamma \Rightarrow \neg\varphi(\exists Q)}{\Gamma \Rightarrow \neg\varphi[\exists Q/\forall Q]}$$

5.3. Additional definitions.

DEFINITION 5.3.1 (Consistency). *A set Γ of formulas is* inconsistent *if there is some formula φ such that both $\Gamma \vdash_{\mathcal{L}} \varphi$ and $\Gamma \vdash_{\mathcal{L}} \neg\varphi$; if there is no such formula φ, the set Γ is* consistent.

By the derived rule (EFQ), a set Γ of formulas is inconsistent iff every formula is deducible in $\mathcal{L}$ from Γ.

DEFINITION 5.3.2 (Height). *The* height *of a deduction D is its height as a tree, i.e., the maximal length of a branch of D.*

§6. Soundness and completeness. We start with the statement, without proof, of several elementary propositions. Soundness and completeness theorems will follow these.

FACT 6.1 (Extensionality). *Suppose a formula φ' in a language $\mathcal{L}$ results from another formula φ in $\mathcal{L}$ on replacing some occurrences of individual constants by some other individual constants of $\mathcal{L}$. Suppose also that an interpretation $\mathcal{M}'$ for $\mathcal{L}$ interprets every constant and predicate of φ' in the same way an interpretation $\mathcal{M}$ interprets the respective symbol in φ. Then: $\varphi'^{\mathcal{M}'} = \varphi^{\mathcal{M}}$.*

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

FACT 6.2. *If c does not occur in a formula φ, then the truth value of φ in $\mathcal{M}_c^\alpha$ is the same as in $\mathcal{M}$.*

FACT 6.3. *If a language $\mathcal{L}_2$ is obtained from $\mathcal{L}_1$ on adding new constants, $\mathcal{M}$ is an interpretation for $\mathcal{L}_2$, and φ is a formula in $\mathcal{L}_1$, then*:

1. *The restriction of $\mathcal{M}$ (as a function) to expressions of $\mathcal{L}_1$ is an interpretation for $\mathcal{L}_1$;*
2. *φ is true in the restriction of $\mathcal{M}$ to $\mathcal{L}_1$ iff it is true in $\mathcal{M}$.*

FACT 6.4. *If $\mathcal{M}$ is an interpretation for a language $\mathcal{L}$, Q is a (possibly compound) one-place predicate of $\mathcal{L}$, and c is an individual constant of $\mathcal{L}$, then*: $[(c)Q]^{\mathcal{M}} = \mathrm{T}$ *iff* $c^{\mathcal{M}} \in Q^{\mathcal{M}}$.

FACT 6.5. *If D is a deduction in $\mathcal{L}$ whose end sequent is $\Gamma \Rightarrow \varphi$, then there is a deduction D' in $\mathcal{L}$ satisfying the following conditions*:

1. *The end sequent of D' is $\Gamma' \Rightarrow \varphi$, where Γ' is a finite subset of Γ.*
2. *Every sequent in D' has finitely many antecedent formulas.*

FACT 6.6. *If a language $\mathcal{L}_2$ results from $\mathcal{L}_1$ on adding new individual constants, then any sequent in $\mathcal{L}_1$ is deducible in $\mathcal{L}_1$ iff it is deducible in $\mathcal{L}_2$.*

FACT 6.7. *If $\Gamma \vdash_{\mathcal{L}} \varphi$ and $\Gamma \subseteq \Delta$, then $\Delta \vdash_{\mathcal{L}} \varphi$.*

FACT 6.8. *If $\Gamma \vdash_{\mathcal{L}} \varphi$, and $\Delta \vdash_{\mathcal{L}} \psi$ for every $\psi \in \Gamma$, then $\Delta \vdash_{\mathcal{L}} \varphi$.*

DEFINITION 6.9 (Valid sequent). *A sequent $\Gamma \Rightarrow \varphi$ in $\mathcal{L}$ is* valid in $\mathcal{L}$ *if $\Gamma \vDash_{\mathcal{L}} \varphi$.*

THEOREM 6.10 (Soundness). *If a sequent is deducible in a language $\mathcal{L}$, then it is valid in $\mathcal{L}$.*

Proof. Straightforward, using induction on the height of deductions. □

THEOREM 6.11 (Completeness). *If a sequent is valid in $\mathcal{L}$, then it is deducible in $\mathcal{L}$.*

The proof of Theorem 6.11 is in the style of (Kearns, 1979) and (Aoyama, 1994). We first note that Theorem 6.11 is an immediate consequence of the following lemma:

LEMMA 6.12. *If $\Gamma \nvdash_{\mathcal{L}} \alpha$, then there is an interpretation for $\mathcal{L}$ in which Γ is true but α is not.*

To prove this lemma, we show that sets with certain properties are 'model-inducing': every set Σ with those properties has a model $\mathcal{M}$ such that, for every formula φ in the language of Σ we have:

1. $\varphi^{\mathcal{M}} = \mathrm{T}$ iff $\varphi \in \Sigma$.
2. $\varphi^{\mathcal{M}} = \mathrm{F}$ iff $\neg\varphi \in \Sigma$.

We then show that, if $\Gamma \nvdash_{\mathcal{L}} \alpha$, then Γ can be extended into a model-inducing set Σ with $\alpha \notin \Sigma$.

We will use the following definitions, facts, and propositions:

DEFINITION 6.13. *A set Σ of formulas in a language $\mathcal{L}$ is* closed under deducibiliy in $\mathcal{L}$ *if it contains all the formulas deducible from it in $\mathcal{L}$.*

DEFINITION 6.14 (Closure under deducibility). *The* closure under deducibility in $\mathcal{L}$ *of a set Σ of $\mathcal{L}$-formulas is the set $Cl_{\mathcal{L}}(\Sigma) = \{\varphi : \Sigma \vdash_{\mathcal{L}} \varphi\}$.*

FACT 6.15. *If Σ is any set of formulas in $\mathcal{L}$, then $Cl_{\mathcal{L}}(\Sigma)$ is closed under deducibility in $\mathcal{L}$ and includes Σ as a subset.*

DEFINITION 6.16 (Henkin set). *A set Σ of formulas in $\mathcal{L}$ is a* Henkin set (*or just* Henkin, *for short*) *if it satisfies the following conditions for every one-place predicate Q*:

1. *If $\varphi(\exists Q) \in \Sigma$, then $(c)Q, \varphi(c) \in \Sigma$ for at least one individual constant c. ($\varphi(c)$ here is $\varphi[\exists Q/c]$.)*
2. *If $\neg\varphi(\forall Q) \in \Sigma$, then $(c)Q, \neg\varphi(c) \in \Sigma$ for at least one individual constant c.*

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

PROPOSITION 6.17. *If a set Σ of $\mathcal{L}$-formulas is Henkin and closed under deducibility in $\mathcal{L}$, then, for every one-place predicate Q and quantifier q*:

1. *If $\varphi(qQ) \in \Sigma$, then $(c)Q, \varphi(c) \in \Sigma$ for some individual constant c.*
2. *If $\neg\varphi(qQ) \in \Sigma$, then $(c)Q, \neg\varphi(c) \in \Sigma$ for some individual constant c.*

Proof. We prove (1) and (2) for the case where q is $\exists$. The proof for $\forall$ is similar. (1) is immediate from Definition 6.16. For (2), suppose that $\neg\varphi(\exists Q) \in \Sigma$. By the derived inference rule $(\exists/\forall)$, we have $\neg\varphi(\exists Q) \vdash_{\mathcal{L}} \neg\varphi(\forall Q)$. Since $\neg\varphi(\exists Q) \in \Sigma$ and Σ is closed under deducibility in $\mathcal{L}$, it follows that $\neg\varphi(\forall Q) \in \Sigma$. Hence, by Definition 6.16, $(c)Q, \neg\varphi(c) \in \Sigma$ for some individual constant c. □

DEFINITION 6.18 (Prime set). *A set Σ of formulas is* prime *if it satisfies the following condition*: *if $\varphi \vee \psi \in \Sigma$, where φ and ψ are formulas without any anaphors of individual constants, then either $\varphi \in \Sigma$ or $\psi \in \Sigma$.*

DEFINITION 6.19 (The counterexample property). *A set Σ of formulas in $\mathcal{L}$* has the counterexample property *if the following conditions hold*:

1. *If $\varphi(\forall Q) \notin \Sigma$, then*: *either $(c)Q \notin \Sigma$ for every individual constant c of $\mathcal{L}$, or there is a constant d of $\mathcal{L}$ such that $(d)Q \in \Sigma$ but $\varphi(d) \notin \Sigma$.*
2. *If $\neg\varphi(\exists Q) \notin \Sigma$, then*: *either $(c)Q \notin \Sigma$ for every individual constant c, or there is a constant d such that $(d)Q \in \Sigma$ but $\neg\varphi(d) \notin \Sigma$.*[25]

DEFINITION 6.20 (Model-inducing set). *A set Σ of formulas in $\mathcal{L}$ is* model-inducing *if it is* (*i*) *consistent,* (*ii*) *Henkin,* (*iii*) *prime,* (*iv*) *closed under deducibility in $\mathcal{L}$, and* (*v*) *has the counterexample property.*

LEMMA 6.21. *Let Σ be a set of formulas in a language $\mathcal{L}$. If Σ is model-inducing, then there is an interpretation $\mathcal{M}$ for $\mathcal{L}$ such that, for every formula φ in $\mathcal{L}$*:

1. $\varphi^{\mathcal{M}} = \mathrm{T}$ *iff* $\varphi \in \Sigma$;
2. $\varphi^{\mathcal{M}} = \mathrm{F}$ *iff* $\neg\varphi \in \Sigma$;
3. $\varphi^{\mathcal{M}} = \mathrm{U}$ *iff* $\varphi, \neg\varphi \notin \Sigma$.

Proof. Condition (3) is entailed by (1) and (2). It is thus sufficient to construct an interpretation $\mathcal{M}$ satisfying (1) and (2). To obtain such an interpretation, we first define a relation $\sim$ on the set of all individual constants of $\mathcal{L}$ in the following way: $c \sim d$ iff $c = d \in \Sigma$. It is easily verified that $\sim$ is an equivalence relation on the set of all $\mathcal{L}$-constants. For every constant c, let $[c]$ be the equivalence class under $\sim$ of c. Since Σ is closed under deducibility in $\mathcal{L}$, we have the following:

> For every n-place predicate R, with $n \geq 1$, and any constants $c_1, \ldots, c_n$, $d_1, \ldots, d_n$, if $[c_i] = [d_i]$ for $i = 1, \ldots, n$, then: $(c_1, \ldots, c_n)R \in \Sigma$ iff $(d_1, \ldots, d_n)R \in \Sigma$.

This fact allows the following definition of $\mathcal{M}$:

- If c is an individual constant of $\mathcal{L}$, then $c^{\mathcal{M}} = [c]$.
- If R is an n-place predicate, $n \geq 1$, then $R^{\mathcal{M}} = \{\langle [c_1], \ldots, [c_n]\rangle : (c_1, \ldots, c_n)R \in \Sigma\}$.

It can now be proved, by induction, that conditions (1) and (2) hold, for every $\mathcal{L}$-formula φ, with respect to the interpretation $\mathcal{M}$ just defined. As part of the induction base, consider a basic formula of the form $c = d$. Condition (1) holds for this formula: $[c = d]^{\mathcal{M}} = \mathrm{T} \Leftrightarrow c^{\mathcal{M}} = d^{\mathcal{M}} \Leftrightarrow [c] = [d] \Leftrightarrow c \sim d \Leftrightarrow c = d \in \Sigma$. As for condition (2), note, first, that exactly one of $c = d$,

[25] Condition (1) can be thought of, intuitively, as follows: Σ "thinks $\varphi(\forall Q)$ is not true" (Σ fails to contain $\varphi(\forall Q)$) only if it "thinks the extension of Q is empty" (Σ does not contain any formula of the form $(c)Q$), or "thinks that some element of that extension fails to satisfy $\varphi(x)$". Condition (2) can be intuitively construed in a similar way.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

$\neg(c = d)$ belongs to Σ: this follows from the deducibity in $\mathcal{L}$ of the sequent $\Sigma \Rightarrow (c = d) \vee \neg(c = d)$ (this is, in fact, an initial sequent), and from the fact that Σ is closed under deducibility in $\mathcal{L}$, as well as prime and consistent. We therefore have: $[c = d]^{\mathcal{M}} = \mathrm{F} \Leftrightarrow [c = d]^{\mathcal{M}} \neq \mathrm{T} \Leftrightarrow c = d \notin \Sigma \Leftrightarrow \neg(c = d) \in \Sigma$. Basic formulas of other forms are treated similarly. As for the induction step, we will consider here in some detail just two examples, associated with the syntactic operations O3 (anaphors) and O4 (quantifiers). As our first example, we prove that condition (1) holds for a formula $\varphi = \psi[o_1/c_x, o_2/x, \ldots, o_m/x]$ resulting from a formula ψ by O3. In this case: $\varphi^{\mathcal{M}} = \mathrm{T} \Leftrightarrow \psi^{\mathcal{M}} = \mathrm{T} \Leftrightarrow$ (by the induction hypothesis) $\psi \in \Sigma \Leftrightarrow$ (by the rules (AI) and (AE), and since Σ is closed under deducibility in $\mathcal{L}$) $\varphi \in \Sigma$. As our second example, we prove that the left-to-right direction of condition (1) holds for a formula $\varphi(\forall Q)$, resulting from formulas $\varphi(c)$ and ψ by O4. Suppose that $\varphi(\forall Q)^{\mathcal{M}} = \mathrm{T}$. Suppose for reductio that $\varphi(\forall Q) \notin \Sigma$. Since Σ has the counterexample property, either (i) $(a)Q \notin \Sigma$ for every individual constant a of $\mathcal{L}$, or (ii) there is some constant d of $\mathcal{L}$ such that $(d)Q \in \Sigma$ but $\varphi(d) \notin \Sigma$. In case (i), it is straightforward to verify that $Q^{\mathcal{M}} = \emptyset$; but this contradicts the truth in $\mathcal{M}$ of $\varphi(\forall Q)$. In case (ii), by the induction hypothesis, $[(d)Q]^{\mathcal{M}} = \mathrm{T}$ and $\varphi(d)^{\mathcal{M}} \neq \mathrm{T}$. Let x be a variable not in $\varphi(\forall Q)$. By Fact 6.4, $[d] \in Q^{\mathcal{M}}$; and we can also see that $\varphi(x)$ is not true of $[d]$ in $\mathcal{M}$. This, again, contradicts the truth in $\mathcal{M}$ of $\varphi(\forall Q)$. □

Proof of Lemma 6.12. Suppose that $\Gamma \nvdash_{\mathcal{L}} \alpha$. We prove that there is an interpretation $\mathcal{M}$ for $\mathcal{L}$ in which every element of Γ is true but α is not. We start by introducing denumerably many new individual constants:

$$c_{0,0,0}, c_{0,0,1}, c_{0,0,2}, \ldots; c_{0,1,0}, c_{0,1,1}, c_{0,1,2}, \ldots; c_{0,2,0}, c_{0,2,1}, c_{0,2,2}, \ldots; \ldots$$
$$c_{1,0,0}, c_{1,0,1}, c_{1,0,2}, \ldots; c_{1,1,0}, c_{1,1,1}, c_{1,1,2}, \ldots; c_{1,2,0}, c_{1,2,1}, c_{1,2,2}, \ldots; \ldots$$
$$c_{2,0,0}, c_{2,0,1}, c_{2,0,2}, \ldots; c_{2,1,0}, c_{2,1,1}, c_{2,1,2}, \ldots; c_{2,2,0}, c_{2,2,1}, c_{2,2,2}, \ldots; \ldots$$
$$\ldots$$

Let $\mathcal{L}_{0,0} = \mathcal{L} \cup \{c_{0,0,0}, c_{0,0,1}, c_{0,0,2}, \ldots\}$. For every $n \in \omega$, let $\mathcal{L}_{0,n+1} = \mathcal{L}_{0,n} \cup \{c_{0,n+1,0}, c_{0,n+1,1}, c_{0,n+1,2}, \ldots\}$. Let $\mathcal{L}_{0,\omega} = \bigcup_{n\in\omega} \mathcal{L}_{0,n}$. Let $\mathcal{L}_{1,0} = \mathcal{L}_{0,\omega} \cup \{c_{1,0,0}, c_{1,0,1}, c_{1,0,2}, \ldots\}$, and let $\mathcal{L}_{1,n+1} = \mathcal{L}_{1,n} \cup \{c_{1,n+1,0}, c_{1,n+1,1}, c_{1,n+1,2}, \ldots\}$. Let $\mathcal{L}_{1,\omega} = \bigcup_{n\in\omega} \mathcal{L}_{1,n}$. We similarly define $\mathcal{L}_{2,\omega}, \mathcal{L}_{3,\omega}, \mathcal{L}_{4,\omega}, \ldots$, and then: $\mathcal{L}_{\omega,\omega} = \bigcup_{n\in\omega} \mathcal{L}_{n,\omega}$. In what follows, by 'the new constants of $\mathcal{L}_{m,n}$' we will mean all the constants of $\mathcal{L}_{m,n}$ that do not belong to any of the preceding languages.

We assume there to be a fixed enumeration of the symbols and formulas of $\mathcal{L}_{\omega,\omega}$, which also yields (on deleting the relevant symbols and formulas) an enumeration of the symbols and formulas of $\mathcal{L}_{n,\omega}$, for each $n \in \omega$, and also of $\mathcal{L}_{m,n}$, for every n and m. We extend Γ into a model-inducing set by performing the following two steps, each of which possibly involves the addition of infinitely many formulas.

- *Step 1.* We first define an increasing sequence $\Gamma_{0,0,0} \subseteq \Gamma_{0,0,1} \subseteq \Gamma_{0,0,2} \subseteq \ldots$ of sets of formulas of $\mathcal{L}_{0,0}$. The set $\Gamma_{0,0,0}$ is defined as the closure of Γ under deducibility in $\mathcal{L}$. We then proceed by induction: for every $n \in \omega$, we let $\Gamma_{0,0,n+1}$ be the set obtained from $\Gamma_{0,0,n}$ in the following way, where γ_{n+1} is the $(n+1)$th formula in $\mathcal{L}_{0,0}$:

  (i) If γ_{n+1} belongs to $\Gamma_{0,0,n}$ and has the form $\varphi(\exists Q)$, then $\Gamma_{0,0,n+1} = \Gamma_{0,0,n} \cup \{(c)Q, \varphi(c)\}$, where c is the first new constant of $\mathcal{L}_{0,0}$ that does not occur in any element of $\Gamma_{0,0,n}$.

  (ii) If γ_{n+1} belongs to $\Gamma_{0,0,n}$ and has the form $\neg\varphi(\forall Q)$, then $\Gamma_{0,0,n+1} = \Gamma_{0,0,n} \cup \{(c)Q, \neg\varphi(c)\}$, where c is the first new constant of $\mathcal{L}_{0,0}$ that does not occur in any element of $\Gamma_{0,0,n}$.

  (iii) If γ_{n+1} belongs to $\Gamma_{0,0,n}$ and has the form $\varphi \vee \psi$, where neither φ nor ψ contains any anaphors of individual constants, then:

$$\Gamma_{0,0,n+1} = \begin{cases} \Gamma_{0,0,n} \cup \{\varphi\} & \text{if } \Gamma_{0,0,n}, \varphi \nvdash_{\mathcal{L}_{0,0}} \alpha, \\ \Gamma_{0,0,n} \cup \{\psi\} & \text{otherwise.} \end{cases}$$

  (iv) If γ_{n+1} does not belong to $\Gamma_{0,0,n}$, or if it is not of any of the forms specified above, then $\Gamma_{0,0,n+1} = \Gamma_{0,0,n}$.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

Let $\Gamma_{0,0,\omega} = \bigcup_{n\in\omega} \Gamma_{0,0,n}$. This is a set of formulas of $\mathcal{L}_{0,0}$. Let $\Gamma_{0,1,0}$ be the closure of $\Gamma_{0,0,\omega}$ under deducibility in $\mathcal{L}_{0,0}$.

Starting with $\Gamma_{0,1,0}$, we repeat the procedure (i)–(iv), this time with respect to the language $\mathcal{L}_{0,1}$. We thus obtain a sequence $\Gamma_{0,1,0}, \Gamma_{0,1,1}, \Gamma_{0,1,2}, \ldots$. Let $\Gamma_{0,1,\omega} = \bigcup_{n\in\omega} \Gamma_{0,1,n}$, and let $\Gamma_{0,2,0}$ be the closure of $\Gamma_{0,1,\omega}$ under deducibility in $\mathcal{L}_{0,1}$.

We repeat this process to obtain an infinite sequence $\Gamma_{0,0,0}, \Gamma_{0,1,0}, \Gamma_{0,2,0}, \ldots$. We conclude Step 1 by defining $\Pi_0 = \bigcup_{n\in\omega} \Gamma_{0,n,0}$. Since each $\Gamma_{0,i,0}$ is a set of formulas of $\mathcal{L}_{0,i-1}$, Π_0 is a set of formulas of $\mathcal{L}_{0,\omega}$.

- *Step 2.* We Now move to the language $\mathcal{L}_{1,0}$. We start by correlating formulas of $\mathcal{L}_{0,\omega}$ with new constants of $\mathcal{L}_{1,0}$. To do this, we go over all the formulas of $\mathcal{L}_{0,\omega}$, in their order in the enumeration of the formulas of that language, and:

  > If the ith formula is $\varphi(\forall Q)$ or $\neg\varphi(\exists Q)$, we correlate the formula with the constant $c_{1,0,2i}$; we call $c_{1,0,2i}$ the *distinguished constant* in $\mathcal{L}_{1,0}$ of that formula (we thus ensure that denumerably many constants of $\mathcal{L}_{0,\omega}$ are not distinguished constants of any $\mathcal{L}_{1,0}$-formula).

  We now define:

$$
\begin{aligned}
B_1 &= \{\beta : \beta \text{ is a formula in } \mathcal{L}_{0,\omega} \text{ and } \beta \notin \Pi_0\}\\
C_1 &= \{\varphi(c^*) : \varphi(\forall Q) \in B_1; \text{ for some constant } c \text{ of } \mathcal{L}_{0,\omega}, (c)Q \in \Pi_0;\\
&\qquad \text{and } c^* \text{ is the distinguished constant in } \mathcal{L}_{1,0} \text{ of } \varphi(\forall Q)\} \ \cup\\
&\quad \{\neg\varphi(c^*) : \neg\varphi(\exists Q) \in B_1; \text{ for some constant } c \text{ of } \mathcal{L}_{0,\omega}, (c)Q \in \Pi_0;\\
&\qquad \text{and } c^* \text{ is the distinguished constant in } \mathcal{L}_{1,0} \text{ of } \neg\varphi(\exists Q)\}\\
A_1 &= \text{the closure of } B_1 \cup C_1 \text{ under disjunction.}
\end{aligned}
$$

  Let $\Gamma_{1,0,0}$ be the set resulting from Π_0 on adding all the formulas $(c^*)Q$ where c^* is the distinguished constant in $\mathcal{L}_{1,0}$ either of a formula $\varphi(\forall Q) \in B_1$ or of a formula $\neg\varphi(\exists Q) \in B_1$, and $(c)Q \in \Pi_0$ for some constant c of $\mathcal{L}_{0,\omega}$. Let $\Gamma_{1,0,n+1}$ be the set obtained from $\Gamma_{1,0,n}$ in the following way, where γ_{n+1} is the $(n+1)$th formula in $\mathcal{L}_{1,0}$ (the following (i)–(iv) resemble the corresponding clauses of Step 1 above, but—unlike the latter—involve the set A_1):

  (i) If γ_{n+1} belongs to $\Gamma_{1,0,n}$ and has the form $\varphi(\exists Q)$, then $\Gamma_{1,0,n+1} = \Gamma_{1,0,n} \cup \{(c)Q, \varphi(c)\}$, where c is the first new constant of $\mathcal{L}_{1,0}$ that does not occur in any element of $\Gamma_{1,0,n} \cup A_1$. (It is easy to verify that such a constant has to exist.)

  (ii) If γ_{n+1} belongs to $\Gamma_{1,0,n}$ and has the form $\neg\varphi(\forall Q)$, then $\Gamma_{1,0,n+1} = \Gamma_{1,0,n} \cup \{(c)Q, \neg\varphi(c)\}$, where c is the first new constant of $\mathcal{L}_{1,0}$ that does not occur in any element of $\Gamma_{1,0,n} \cup A_1$.

  (iii) If γ_{n+1} belongs to $\Gamma_{1,0,n}$ and has the form $\varphi \vee \psi$, where neither φ nor ψ contains anaphors of constant occurrences, then:

$$
\Gamma_{1,0,n+1} = \begin{cases} \Gamma_{1,0,n} \cup \{\varphi\} & \text{if } \Gamma_{1,0,n}, \varphi \vdash_{\mathcal{L}_{1,0}} \gamma \text{ for every } \gamma \in A_1,\\ \Gamma_{1,0,n} \cup \{\psi\} & \text{otherwise.} \end{cases}
$$

  (iv) If γ_{n+1} does not belong to $\Gamma_{1,0,n}$, or if it is not of any of the forms specified above, then $\Gamma_{1,0,n+1} = \Gamma_{1,0,n}$.

  Let $\Gamma_{1,0,\omega} = \bigcup_{n\in\omega} \Gamma_{1,0,n}$, and let $\Gamma_{1,1,0}$ be the closure of $\Gamma_{1,0,\omega}$ under deducibility in $\mathcal{L}_{1,0}$. We now move to the next language, $\mathcal{L}_{1,1}$, and repeat the procedure (i)–(iv) above to obtain $\Gamma_{1,1,1}, \Gamma_{1,1,2}, \Gamma_{1,1,3}, \ldots$. Similarly to what we did before, we take $\Gamma_{1,1,\omega} = \bigcup_{n\in\omega} \Gamma_{1,1,n}$, and define $\Gamma_{1,2,0}$ as the closure of $\Gamma_{1,1,\omega}$ under deducibility in $\mathcal{L}_{1,1}$. We continue in the same way indefinitely, to obtain the sequence $\Gamma_{1,0,0}, \Gamma_{1,1,0}, \Gamma_{1,2,0}, \Gamma_{1,3,0}, \Gamma_{1,4,0}, \ldots$. Each $\Gamma_{1,i,0}$ is a set of formulas of $\mathcal{L}_{1,i-1}$. Let $\Pi_1 = \bigcup_{n\in\omega} \Gamma_{1,n,0}$. Π_1 is a set of formulas of $\mathcal{L}_{1,\omega}$.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

Step 2, by which we obtained Π_1 is now repeated indefinitely, to obtain Π_2, Π_3, Π_4, ... Every Π_i is a set of formulas in $\mathcal{L}_{i,\omega}$. Finally, we define $\Pi = \bigcup_{n\in\omega} \Pi_n$. Π is a set of formulas of $\mathcal{L}_{\omega,\omega}$.

We obviously have $\Gamma \subseteq \Pi$. It will now suffice to show that Π is model-inducing and that $\alpha \notin \Pi$. (This, together with Lemma 6.21 and Fact 6.3, will entail that there is an interpretation for $\mathcal{L}$ in which Γ is true but α is not.) The required result follows from the next three claims:

Claim 1. Each Π_n, as well as Π, is prime and Henkin.

Claim 2. Each Π_n is closed under deducibility in $\mathcal{L}_{n,\omega}$, and Π is closed under deducibility in $\mathcal{L}_{\omega,\omega}$.

Claim 3. Π is consistent, has the counterexample property, and $\alpha \notin \Pi$.

The proofs of claims (1) and (2) are straightforward. The proof of (3) relies on three additional claims:

Claim 3.1. For each $n \in \omega$ and $\beta \in A_1$, the sequent $\Gamma_{1,0,n} \Rightarrow \beta$ is not deducible in $\mathcal{L}_{1,0}$.

Claim 3.2. For each $n \in \omega$ and $\beta \in A_{n+1}$, the sequent $\Pi_{n+1} \Rightarrow \beta$ is not deducible in $\mathcal{L}_{n+1,\omega}$.

Claim 3.3. $\alpha \in A_1 \subseteq A_2 \subseteq \ldots$.

We omit the proofs of claims (3.1)–(3.3), as well as the detailed proof of claim (3). □

§7. Quarc and PC. This section treats the translation of PC into Quarc and vice versa. We first show, in §7.1, that PC-sentences can be straightforwardly translated into Quarc, and that the translation preserves truth-in-an-interpretation and falsity-in-an-interpretation, as well as entailment. Later, in §7.2, we show that, although an entailment-preserving translation of Quarc into PC is possible, no such translation can be straightforward, in a sense that we make clear, and which is applicable to the PC-to-Quarc translation of §7.1.

7.1. PC-to-Quarc translation. We will start by presenting the basic idea employed in the translation of PC into Quarc; a more precise treatment will then follow. Quantification in PC is always in one of the following forms:

$$\forall x \varphi(x). \tag{1}$$

$$\exists x \varphi(x). \tag{2}$$

Intuitively, these can be thought of as 'everything is φ' and 'something is φ', respectively; or:

$$\text{Everything is a thing } x \text{ such that } \varphi(x) \tag{3}$$

$$\text{Something is a thing } x \text{ such that } \varphi(x). \tag{4}$$

Now, (3) and (4) can be straightforwardly translated into Quarc: representing 'thing' by a one-place predicate T, we can translate 'Everything' and 'Something' as the quantified arguments $\forall T$ and $\exists T$;[26] (3) and (4) can then be translated, respectively, as:

$$(\forall T)\ \ T_x : \varphi(x) \tag{5}$$

$$(\exists T)\ \ T_x : \varphi(x). \tag{6}$$

Under a suitable restriction on the extension of T ('thing'), the semantics of (5) and (6) becomes essentially the same as that of (1) and (2). The restriction is this: the extension of T should never be empty, and should, moreover, contain the denotations of all individual constants.

To make this more rigorous, we will define a translation function from the formulas of PC to those of Quarc. We will then correlate models of a certain axiom scheme—a scheme expressing the above restriction on the extension of T—with interpretations of PC.

[26] We are thus treating 'everything' and 'something' as equivalent to 'every thing' and 'some thing', respectively. Comparison to analogs in other languages of 'everything' and of 'something', as well as to other quantified expressions in English, such as 'two things' and 'many things', supports the plausibility of this treatment. I am indebted to an anonymous referee for this point.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

The version of PC that will be employed here is one with identity and without function symbols. 'PC' will henceforth be used as a name for that specific version. Any language $\mathcal{L}_P$ of PC contains denumerably many individual constants, and may also contain finitely or denumerably many n-place predicates for each $n \geq 1$. The *atomic formulas* of $\mathcal{L}_P$ are all the expressions of the forms $c_1 = c_2$ and $R(c_1, \ldots, c_n)$, where c_i are individual constants and where R is an n-place predicate, $n \geq 1$. The set of *formulas* of $\mathcal{L}_P$ is the closure of the set of atomic formulas under negation, disjunction, conjunction, and implication, and under the operation $\varphi(c) \mapsto \forall x\varphi[c/x], \exists x\varphi[c/x]$, where c is an individual constant, x is a variable that does not occur in $\varphi(c)$, and $\varphi[c/x]$ results from $\varphi(c)$ on replacing all the occurrences of c with x.

An *interpretation* $\mathcal{M}$ for $\mathcal{L}_P$ is defined in the usual way. Every interpretation $\mathcal{M}$ for $\mathcal{L}_P$ has a nonempty set $|\mathcal{M}|$ as its domain. An interpretation $\mathcal{M}$ assigns an element $c^{\mathcal{M}}$ of $|\mathcal{M}|$ to every individual constant c of $\mathcal{L}_P$, and a subset $R^{\mathcal{M}}$ of $|\mathcal{M}|^n$ to every n-place predicate R, $n \geq 1$. The truth-value, T or F, of atomic formulas in $\mathcal{M}$ is defined in the usual way; truth-functional compounds are treated classically. If $\alpha \in |\mathcal{M}|$, then $\mathcal{M}_c^\alpha$ is the interpretation for $\mathcal{L}_P$ obtained from $\mathcal{M}$ on replacing $c^{\mathcal{M}}$ with α. And an element α of $|\mathcal{M}|$ *satisfies* $\varphi[c/x]$ in $\mathcal{M}$ if $\varphi(c)$ is true in $\mathcal{M}_c^\alpha$. The truth conditions of quantified formulas are now defined as follows: $[\forall x\varphi[c/x]]^{\mathcal{M}} = \mathrm{T}$ if all the elements of $|\mathcal{M}|$ satisfy $\varphi[c/x]$ in $\mathcal{M}$, and F otherwise. $[\exists x\varphi[c/x]]^{\mathcal{M}} = \mathrm{T}$ if at least one element of $|\mathcal{M}|$ satisfies $\varphi[c/x]$ in $\mathcal{M}$, and F otherwise. Entailment is defined as usual.

DEFINITION 7.1.1 (Language-correlation). *With each PC-language $\mathcal{L}_P$ we correlate the unique Quarc-language $\mathcal{L}_Q$ that satisfies the following requirements*:

1. *The individual constants of $\mathcal{L}_Q$ are the ones of $\mathcal{L}_P$.*
2. *For each $n > 1$, the n-place predicates of $\mathcal{L}_Q$ are those of $\mathcal{L}_P$.*
3. *The one-place predicates of $\mathcal{L}_Q$ are those of $\mathcal{L}_P$ together with one additional one-place predicate T. (Intuitively, T will stand for 'thing'.)*

DEFINITION 7.1.2 (Translation). *The translation $t(\varphi)$ of a formula φ in $\mathcal{L}_P$ is defined as follows (by induction on the complexity of φ)*:

1. *Atomic formulas*:

$$t(R(c_1, \ldots, c_n)) = (c_1, \ldots, c_n)R$$
$$t(c_1 = c_2) = c_1 = c_2.$$

2. *Truth functional compounds*:

$$t(\neg\varphi) = \neg t(\varphi)$$
$$t(\varphi * \psi) = t(\varphi) * t(\psi), \text{ where } * \text{ is } \vee, \wedge, \text{ or } \rightarrow.$$

3. *Quantifiers*:

$$t(\forall x\varphi[c/x]) = (\forall T)\ T_x : t(\varphi)[c/x]$$
$$t(\exists x\varphi[c/x]) = (\exists T)\ T_x : t(\varphi)[c/x].$$

DEFINITION 7.1.3 (CDT). *CDT (constants denote things) is the set of all formulas of the form $(c)T$, where c is an individual constant of $\mathcal{L}_Q$.*

We will now associate PC-interpretations with (Quarc-) models of CDT. The translation t will be shown to preserve truth-values in the following sense: a PC-formula φ will be true (false) in a given PC-interpretation iff its Quarc-translation, $t(\varphi)$, is true (false) in the associated Quarc-interpretation. Entailment will be preserved in the following way: the entailment of an $\mathcal{L}_P$-formula φ by a set Γ of $\mathcal{L}_P$-formulas will be equivalent to the entailment of $t(\varphi)$ by $t(\Gamma)$ and CDT.

DEFINITION 7.1.4 (Induced interpretation). *Let $\mathcal{M}$ be a (Quarc) model of CDT. The interpretation* induced by *$\mathcal{M}$ is the unique interpretation $\mathcal{M}^*$ for $\mathcal{L}_P$ that satisfies the following requirements*:

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

1. $|\mathcal{M}^*| = T^{\mathcal{M}}$.
2. $c^{\mathcal{M}^*} = c^{\mathcal{M}}$ *for every individual constant* c.
3. $R^{\mathcal{M}^*} = R^{\mathcal{M}} \cap (T^{\mathcal{M}})^n$ *for every* n*-place predicate* R $(n \geq 1)$ *of* $\mathcal{L}_P$.

For the rest of this subsection we will use the term '$\mathcal{M}^*$' as designating the interpretation induced by $\mathcal{M}$.

FACT 7.1.5. *If* c *and* d *are any individual constants of* $\mathcal{L}_Q$ *(and thus also of* $\mathcal{L}_P$*) and* $\varphi(c)$ *is a formula of* $\mathcal{L}_P$*, then* $[t(\varphi)][c/d] = t(\varphi[c/d])$.

FACT 7.1.6. *If* $\mathcal{M}$ *is a model of CDT and* $\alpha \in T^{\mathcal{M}}$*, then*:

1. $\mathcal{M}_c^\alpha$ *is a model of CDT.*
2. $[\mathcal{M}_c^\alpha]^* = [\mathcal{M}^*]_c^\alpha$.

FACT 7.1.7. *Every interpretation for* $\mathcal{L}_P$ *is induced by some model of CDT.*

THEOREM 7.1.8 (Invariance of truth-values under t). *If* φ *is a formula in* $\mathcal{L}_P$ *and* $\mathcal{M}$ *is a model of CDT, then* $[t(\varphi)]^{\mathcal{M}} = [\varphi]^{\mathcal{M}^*}$.

Proof. By induction on the complexity of φ. □

COROLLARY 7.1.9. *Let* φ *be a formula of* $\mathcal{L}_P$*, and let* $\mathcal{M}$, $\mathcal{M}'$ *be models of CDT. Then*:

1. $[t(\varphi)]^{\mathcal{M}}, [t(\varphi)]^{\mathcal{M}'} \in \{T, F\}$
2. *If* $\mathcal{M}^* = \mathcal{M}'^*$*, then* $[t(\varphi)]^{\mathcal{M}} = [t(\varphi)]^{\mathcal{M}'}$.

Proof. Clause (1) is immediate from Theorem 7.1.8. For (2), note that $[t(\varphi)]^{\mathcal{M}} = [\varphi]^{\mathcal{M}^*} = [\varphi]^{\mathcal{M}'^*} = [t(\varphi)]^{\mathcal{M}'}$. □

Clause (2) of the last Corollary means that translations into $\mathcal{L}_Q$ of $\mathcal{L}_P$-formulas cannot distinguish between any two models of CDT that induce the same interpretation for $\mathcal{L}_P$. This, in turn, means that, whenever the discussion is restricted to translations of $\mathcal{L}_P$-formulas, what is essential is not whole Quarc-interpretations, but only the PC-interpretations that they induce.

THEOREM 7.1.10 (Invariance of entailment under t). *Let* $\Gamma \cup \{\varphi\}$ *be a set of formulas in* $\mathcal{L}_P$*. Then:* $\Gamma \vDash \varphi$ *iff* $t(\Gamma), CDT \vDash t(\varphi)$.

Proof. Suppose, first, that $\Gamma \nvDash \varphi$. Then there is a model $\mathcal{M}'$ of Γ in which φ is not true. By Fact 7.1.7, there is a (Quarc-) model $\mathcal{M}$ of CDT such that $\mathcal{M}^* = \mathcal{M}'$. By Theorem 7.1.8, $[t(\psi)]^{\mathcal{M}} = [\psi]^{\mathcal{M}^*}$ for every $\psi \in \Gamma \cup \{\varphi\}$. Thus, $\mathcal{M}$ is a model of $t(\Gamma) \cup$ CDT, but not of $t(\varphi)$. Hence, $t(\Gamma)$, CDT $\nvDash t(\varphi)$, as required. Suppose, conversely, that $t(\Gamma)$, CDT $\nvDash t(\varphi)$. Then, there is a model $\mathcal{M}$ of $t(\Gamma) \cup CDT$ in which $t(\varphi)$ is not true. By Theorem 7.1.8, $[t(\psi)]^{\mathcal{M}} = [\psi]^{\mathcal{M}^*}$ for every $\psi \in \Gamma \cup \{\varphi\}$. The interpretation $\mathcal{M}^*$ is thus a model of Γ in which φ is not true. Hence, $\Gamma \nvDash \varphi$. □

7.2. Quarc-to-PC translation. The translation of PC into Quarc defined in the previous subsection is straightforward in the following sense: it can be obtained by translating PC-sentences into English in a standard way, and then formalizing the resulting English translations in Quarc, again—in a standard way. I will not define 'standard' here. I submit, however, that there is an intuitive understanding of that notion, on which: (i) the above holds; and (ii) any standard translation from PC or Quarc to English (or vice versa) involves the translation of $\neg$, $\wedge$, $\vee$, and $\rightarrow$ by (respectively) 'it is not the case that', 'and', 'or', and 'if... then...' (or vice versa).

It is perhaps natural to ask whether a similar translation is possible in the opposite direction—namely, from Quarc to PC. Obviously, the truth-values of Quarc-formulas cannot in general be preserved by any translation into PC; for Quarc is a three-valued logic while PC is two-valued. We can still ask, however, if there is a straightforward translation from Quarc to PC that preserves the truth-value T and entailment. As I will explain below: (i) there is a translation from Quarc to PC that preserves the truth-value T, as well as entailment; however, (ii) no such translation can be straightforward in the sense specified above.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

To construct a translation as described in (i), we begin, as before, with a correlation between Quarc- and PC-languages. The correlation will be slightly different from that of §7.1; for we will now not need to designate a special predicate for 'thing':

DEFINITION 7.2.1 (Language-correlation). *With each Quarc language $\mathcal{L}_Q$ we correlate the unique PC-language $\mathcal{L}_P$ for which*:

1. *The individual constants of $\mathcal{L}_P$ are the ones of $\mathcal{L}_Q$.*
2. *For every n, the n-place predicates of $\mathcal{L}_P$ are the primitive n-place predicates of $\mathcal{L}_Q$.*

To track the truth of formulas, we will supply, for every Quarc-formula φ, a PC-translation $\hat{t}(\varphi)$, as well as a separate translation $\hat{t}(\neg\varphi)$ of $\neg\varphi$, which will in general be distinct from $\neg\hat{t}(\varphi)$. The formula $\hat{t}(\varphi)$ can be thought of as expressing the truth-conditions of φ, while $\hat{t}(\neg\varphi)$ can be thought of as expressing its *falsity*-conditions.

DEFINITION 7.2.2 (Translation). *The PC-translation $\hat{t}(\varphi)$ of a formula φ in $\mathcal{L}_Q$ is defined as follows (by induction on the complexity of φ)*:

1. *Basic formulas*:

$$\begin{aligned}
&\hat{t}(c_1 = c_2) = c_1 = c_2\\
&\hat{t}(c_1 \neq c_2) = \neg(c_1 = c_2)\\
&\hat{t}((c_1, \ldots, c_n)R) = R(c_1, \ldots, c_n)\\
&\hat{t}((c_1, \ldots, c_n)\neg R) = \neg R(c_1, \ldots, c_n)\\
&\hat{t}(\neg(c_1 = c_2)) = \neg(c_1 = c_2)\\
&\hat{t}(\neg(c_1 \neq c_2)) = c_1 = c_2\\
&\hat{t}(\neg(c_1, \ldots, c_n)R) = \neg R(c_1, \ldots, c_n)\\
&\hat{t}(\neg(c_1, \ldots, c_n)\neg R) = R(c_1, \ldots, c_n)
\end{aligned}$$

2. *Compound predicates*:

$$\begin{aligned}
&\hat{t}((c)\ P_x : \psi(x)) = \hat{t}((c)P) \wedge \hat{t}(\psi(c))\\
&\hat{t}((c)\ \neg P_x : \psi(x)) = \hat{t}(\neg(c)P) \vee \hat{t}(\neg\psi(c))\\
&\hat{t}(\neg(c)\ P_x : \psi(x)) = \hat{t}(\neg(c)P) \vee \hat{t}(\neg\psi(c))\\
&\hat{t}(\neg(c)\ \neg P_x : \psi(x)) = \hat{t}((c)P) \wedge \hat{t}(\psi(c))
\end{aligned}$$

3. *Truth-functional compounds*:

$$\begin{aligned}
&\hat{t}(\varphi \wedge \psi) = \hat{t}(\varphi) \wedge \hat{t}(\psi)\\
&\hat{t}(\varphi \vee \psi) = \hat{t}(\varphi) \vee \hat{t}(\psi)\\
&\hat{t}(\varphi \rightarrow \psi) = \hat{t}(\neg\varphi) \vee \hat{t}(\psi)\\
&\hat{t}(\neg(\varphi \wedge \psi)) = \hat{t}(\neg\varphi) \vee \hat{t}(\neg\psi)\\
&\hat{t}(\neg(\varphi \vee \psi)) = \hat{t}(\neg\varphi) \wedge \hat{t}(\neg\psi)\\
&\hat{t}(\neg(\varphi \rightarrow \psi)) = \hat{t}(\varphi) \wedge \hat{t}(\neg\psi)\\
&\hat{t}(\neg\neg\varphi) = \hat{t}(\varphi)
\end{aligned}$$

4. *Constant anaphors*: *if φ results from ψ by the operation O3 of §3, then*:

$$\begin{aligned}
&\hat{t}(\varphi) = \hat{t}(\psi)\\
&\hat{t}(\neg\varphi) = \hat{t}(\neg\psi)
\end{aligned}$$

5. *Quantified arguments*:

$$\begin{aligned}
&\hat{t}(\varphi(\forall P)) = \exists x[\hat{t}((x)P)] \wedge \forall x[\hat{t}((x)P) \rightarrow \hat{t}(\varphi(x))]\\
&\hat{t}(\varphi(\exists P)) = \exists x[\hat{t}((x)P) \wedge \hat{t}(\varphi(x))]\\
&\hat{t}(\neg\varphi(\forall P)) = \exists x[\hat{t}((x)P) \wedge \hat{t}(\neg\varphi(x))]\\
&\hat{t}(\neg\varphi(\exists P)) = \exists x[\hat{t}((x)P)] \wedge \forall x[\hat{t}((x)P) \rightarrow \hat{t}(\neg\varphi(x))]
\end{aligned}$$

By $\hat{t}(\varphi(x))$ here we mean $\hat{t}(\varphi[x/c])[c/x]$, where c is a constant that does not occur in $\varphi(x)$, and similarly for $\hat{t}((x)P)$.

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

DEFINITION 7.2.3 (Induced interpretation). *Let $\mathcal{M}$ be an interpretation for $\mathcal{L}_P$. The interpretation* induced by *$\mathcal{M}$ is the unique interpretation $\mathcal{M}^*$ for $\mathcal{L}_Q$ that satisfies the following requirements*:

1. $c^{\mathcal{M}^*} = c^{\mathcal{M}}$ *for every individual constant c.*
2. $R^{\mathcal{M}^*} = R^{\mathcal{M}}$ *for every primitive predicate R, of any arity.*

We will henceforth use '$\mathcal{M}^*$' to designate the interpretation induced by $\mathcal{M}$.

FACT 7.2.4. *Every interpretation for $\mathcal{L}_Q$ is induced by some interpretation for $\mathcal{L}_P$.*

FACT 7.2.5. *If $\mathcal{M}$ is an interpretation for $\mathcal{L}_P$ and $\alpha \in |\mathcal{M}|$, then $[\mathcal{M}^{\alpha}_{c}]^* = [\mathcal{M}^*]^{\alpha}_{c}$.*

THEOREM 7.2.6 (Invariance of truth under $\hat{t}$). *If φ is a formula in $\mathcal{L}_Q$ and $\mathcal{M}$ is an interpretation for $\mathcal{L}_P$, then $[\hat{t}(\varphi)]^{\mathcal{M}} = \mathrm{T}$ iff $[\varphi]^{\mathcal{M}^*} = \mathrm{T}$.*

Proof. To prove this theorem, we prove, by induction on the complexity of φ that (i) $[\hat{t}(\varphi)]^{\mathcal{M}} = \mathrm{T}$ iff $[\varphi]^{\mathcal{M}^*} = \mathrm{T}$ and (ii) $[\hat{t}(\neg\varphi)]^{\mathcal{M}} = \mathrm{T}$ iff $[\neg\varphi]^{\mathcal{M}^*} = \mathrm{T}$. We suppress the details. □

THEOREM 7.2.7 (Invariance of entailment under $\hat{t}$). *Let $\Gamma \cup \{\varphi\}$ be a set of formulas in $\mathcal{L}_Q$. Then $\Gamma \vDash \varphi$ iff $\hat{t}(\Gamma) \vDash \hat{t}(\varphi)$.*

Proof. Similar to that of Theorem 7.1.10. □

The following theorem, with which we close this subsection, ensures that no translation of Quarc into PC is both entailment-preserving and straightforward (in the sense specified above).

THEOREM 7.2.8. *There is no function $\bar{t}$ from the formulas of $\mathcal{L}_Q$ to those of $\mathcal{L}_P$ for which the following conditions hold*:

1. $\bar{t}(\neg\varphi) = \neg\bar{t}(\varphi)$ *for every formula φ in $\mathcal{L}_Q$.*
2. *There is a set Δ of $\mathcal{L}_P$-sentences such that, for every pair φ, ψ of $\mathcal{L}_Q$-formulas, $\bar{t}(\varphi), \Delta \vDash \bar{t}(\psi)$ iff $\varphi \vDash \psi$.*

Proof. Suppose for reductio that there *is* such a function $\bar{t}$. Let $\varphi = (\forall S)P$, and let $\psi = (\exists S)S$. Then:

(*) $\varphi \vDash \psi$ and $\neg\varphi \vDash \psi$;

(**) It is not the case that $\chi \vDash \psi$ for every formula χ in $\mathcal{L}_Q$.

From (*) and (2): $\bar{t}(\neg\varphi), \Delta \vDash \bar{t}(\psi)$. And by (1): $\neg\bar{t}(\varphi), \Delta \vDash \bar{t}(\psi)$. From (*) and (2) it also follows that $\bar{t}(\varphi), \Delta \vDash \bar{t}(\psi)$. Hence, $\Delta \vDash \bar{t}(\psi)$, and so $\bar{t}(\chi), \Delta \vDash \bar{t}(\psi)$ for every $\mathcal{L}_Q$-formula χ. By (2), $\chi \vDash \psi$ for every such χ, contradicting (**). □

§8. Concluding remarks. As was seen in the previous section, the first-order predicate calculus (PC) can be translated into Quarc in much the same way as it is translated into English. The translation was shown to preserve truth-values and entailment. Given these results, the following perspective on PC becomes available. PC can be viewed as a restricted logic: it results from Quarc on limiting quantification to constructions of the forms

$$(\forall T)\, T_x : \varphi(x) \qquad (\text{'Everything is a thing } x \text{ such that } \varphi(x)\text{'}) \tag{1}$$

and

$$(\exists T)\, T_x : \varphi(x) \qquad (\text{'Something is a thing } x \text{ such that } \varphi(x)\text{'}), \tag{2}$$

where T is a particular, fixed-in-advance, one-place predicate, and on limiting the denotations of individual constants to elements of the extension of T.[27] These restrictions, in turn, can be viewed as

[27] We may also disallow the following syntactic devices of Quarc: negation in the form $(c_1, \ldots, c_n)\neg R$, compound predicates in the predicate positions (the restrictions formulated in

https://doi.org/10.1017/S1755020317000053 Published online by Cambridge University Press

akin to simplifying assumptions, and can be motivated by practical considerations:[28] if T is the only predicate to which quantifiers are allowed to attach, and if this predicate is assumed to be nonempty, then truth-value gaps are guaranteed not to occur. This simplifies the semantics considerably, and allows the employment of much simpler proof-systems than the one introduced in §5.

There is, however, a price to pay for the simplicity gained by the above restrictions. First, those restrictions bring with them a considerable measure of prolixity in formalization: a sentence such as

All human beings are mortal (3)

can no longer be represented as

$$(\forall H)M; \tag{4}$$

it now has to be paraphrased, as something like

Everything is a thing x such that, if x is a human being, then x is mortal, (5)

and only then translated, as

$$(\forall T)\, T_x : [(x)H \rightarrow (x)M]. \tag{6}$$

Second, to the extent that the analysis on which Quarc is founded is correct, the rendering of sentences such as (3) in the forms (5) and (6) distorts their semantics, and misrepresents entailment relations between them.[29]

A related consequence of the above restrictions is that the extension of T gains a special logical status: our restricted formulas quantify only over the elements of that extension; and nothing outside that extension is relevant to the truth-values of those formulas in any interpretation (see §7, Corollary 7.1.9). The extension of T thus comes to play the role of the *domain of quantification* of ordinary model-theoretic semantics: a fixed, nonempty, set over which the variables of quantification are allowed to range.

The above restrictions—the ones by which PC results from Quarc—may or may not be desirable, depending on what one is trying to achieve with one's logic. But it is important to note, I think, that those restrictions are *optional*. The syntactic and semantic devices by which quantification is achieved in PC—attachment of quantifiers to open formulas and reference to a presupposed domain—are not necessary components of logic or of model-theory: from the perspective suggested here, this way of achieving quantification is merely a byproduct of the above restrictions. If we choose *not* to impose those restrictions, then we are left with a logic at least as capable as PC (and, arguably, more capable) in representing inferences made in natural language; and we still have a model-theory, as well as a sound and complete proof-system, for this logic. All this may have some interesting implications for philosophical discussions in which the notion of a domain of quantification is assumed; a case in point would be the Quinean approach to ontology. But I leave the discussion of such philosophical implications to another occasion.

§9. Acknowledgements. This paper has greatly benefited from comments on earlier versions by Hanoch Ben-Yami, Eli Dresner, Norbert Gratzl, Naomi Korem, Edi Pavlović, Gil Sagi, Stewart Shapiro, and three anonymous referees for this journal. I have also greatly benefited from discussions with Arnon Avron. Parts of this paper were presented at an advanced logic seminar at the Central

the text already disallow compound predicates as part of quantified arguments), reordered forms, and anaphors other than in the positions displayed in (1) and (2); but I take these additional restrictions to be less important than the ones mentioned in the text: I believe the resulting (restricted) logic can still be counted as a variant of the first-order predicate calculus if we impose only the restrictions mentioned in the text.

[28] This is, of course, *not* how the PC-treatment of natural-language sentences was historically motivated. Frege argued for the now-standard PC-treatment in several places. See (Ben-Yami, 2006) for a critical examination of the Fregean arguments.

[29] See §2.

European University, as well as at a workshop at the Munich Center for Mathematical Philosophy at LMU Munich, and I have benefited considerably from comments I received on both occasions. The research leading to this paper was supported by a Polonsky fellowship at the Hebrew University of Jerusalem.

BIBLIOGRAPHY

Aoyama, H. (1994). The strong completeness of a system based on Kleene's strong three-valued logic. *Notre Dame Journal of Formal Logic*, **35**(3), 355–368.

Barwise, J. & Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, **4**(2), 159–219.

Ben-Yami, H. (2004). *Logic and Natural Language: On Plural Reference and its Semantic and Logical Significance*. Aldershot, Hants: Ashgate.

Ben-Yami, H. (2006). A critique of Frege on common nouns. *Ratio*, **19**(2), 148–155.

Ben-Yami, H. (2014). The quantified argument calculus. *The Review of Symbolic Logic*, **7**(01), 120–146.

Francez, N. (2014). A logic inspired by natural language quantifiers as subnectors. *Journal of Philosophical Logic*, **43**(6), 1153–1172.

Geach, P. T. (1962). *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Ithaca, NY: Cornell University Press.

Groenendijk, J. & Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, **14**(1), 39–100.

Kearns, J. T. (1979). The strong completeness of a system for Kleene's three-valued logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **25**(3–6), 61–68.

Kneale, W. & Kneale, M. (1971). *The Development of Logic*. London: Oxford University Press.

Lanzet, R. & Ben-Yami, H. (2004). Logical inquiries into a new formal system with plural reference. In Hendricks, V., Neuhaus, F., Pedersen, S. A., Scheffler, U., and Wansing, H., editors. *First-Order Logic Revisited*. Berlin: Logos Verlag, pp. 173–223.

Moss, L. S. (2010). Logics for two fragments beyond the syllogistic boundary. In Blass, A., Dershowitz, N., and Reisig, W., editors. *Fields of Logic and Computation: Essays Dedicated to Yuri Gurevich on the Occasion of his 70th Birthday*. Berlin: Springer, pp. 538–564.

Pratt-Hartmann, I. & Moss, L. (2009). Logics for the relational syllogistic. *Review of Symbolic Logic*, **2**(4), 647–683.

Strawson, P. F. (1950). On referring. *Mind*, **59**(235), 320–344.

DEPARTMENT OF PHILOSOPHY
TEL-AVIV UNIVERSITY
P.O. BOX 39040
RAMAT AVIV, ISRAEL
E-mail: lanzetr@gmail.com