Grice on Vacuous Names

Roger Bishop Jones

Abstract

Formal analysis (using Higher Order Logic with ProofPower) and commentary on Grice's system Q (G, G_{HP}) first presented in his paper *Vacuous Names*.

Created 2010/07/07

Last Change Date: 2011/02/02 21:48:21

http://www.rbjones.com/rbjpub/pp/doc/t037.pdf

Id: t037a.tex,v 1.5 2011/02/02 21:48:21 rbj Exp

Contents

1 Prelude

This document is intended possibly to form a chapter of *Analyses of Analysis* [7].

My initial purpose in preparing the document is simply to investigate and analyse Grice's "System Q" (since renamed "System G" by Myro and "System G_{HP}" by Speranza) using a decent proof tool.

Further discussion of what might become of this document in the future may be found in my postscript (Section 9).

In this document, phrases in coloured text are hyperlinks, like on a web page, which will usually get you to another part of this document (the blue parts, the contents list, page numbers in the Index) but sometimes take you (the red bits) somewhere altogether different (if you happen to be online) like the hist-analytic archives.

For description of the formal languages, methods and tools used in or in producing this document see: [6].

2 Introduction

I will endeavour to follow the structure of Grice's paper *Vacuous names*[4, 2]. However, the method employed for the formal analysis, which is called "shallow embedding", is essentially semantic, and involves addressing semantic fundamentals before finding manageable syntactic presentations, and I therefore find it necessary to address key aspects of the semantics before considering the details of syntax.

2.1 Background

Grice's deliberations about Vacuous Names may be understood primarily against those who have supposed or desired a significant difference between formal logics and the logic of ordinary discourse, by one who felt that the differences are, or need only be, modest. In some of his writings. for example in the "Retrospective Epilogue" in *Sudies in the Way of Words*[5], Grice describes two camps, "Modernists" and "Neo-traditionalists", the former making more and the latter less of the distinctions between formal and ordinary logic.

Elsewhere in Grice's philosophy, in connection with this controversy, the relationship between the logical connectives and their ordinary language counterparts is discussed. Here we are concerned with the problem of referring expressions which lack a referent.

Modern discussion of this problem begins with Russell [10, 11], who sought formal languages for mathematics which were more precise and more transparent in their logical structure than is ordinary language.

There are two aspects of the problems here at stake which are separable. One concerns logic and language; how we talk and reason using words or phrases which do not, or might not, refer to any existent entity. The second is the metaphysical question of what exists. Russell's most important work in this area arose from his making a transition away from a lavish Meinongian ontology in which there are different kinds or grades of existence, to the adoption of Occam's razor, leading to a spartan ontology. Key to this is the acceptance that some referring phrases fail to refer, and Russell's principal contribution here is in his theory of descriptions [11, 10].

Russell's theory treats definite and indefinite descriptions as "incomplete symbols", which contribute to the meaning of the sentence as a whole even if they fail to refer. Such incomplete symbols are only meaningful in certain particular context, and their meaning is given by translated the whole into some other phrase in which the incomplete symbol does not occur and in which there remains no explicit reference to an entity which might not exist. Usually this involves quantification. Proper names are then eliminated in favour of descriptions.

Grice's principal aim is to resurrect the conception of names as references, rather than as surrogates for descriptions. He wants to do this without being forced to regard names of non-existent entities, such as Pegasus, as yielding sentences which are neither true nor false. This he achieves by allowing that names which do not designate nevertheless refer to some kind of entity which he calls its correlate, and by the rule that predications to non-designating names are to yield falsehoods.

2.2 Preliminary Formalities

I consider two different approaches, both derived from Grice, to the problems discussed in *Vacuous Names*[4]. Much of the discussion concerns matters which are common to both, and this involves some formal material. In order to avoid replication the formalisation of this common ground is provided first in a separate theory before matters specific to the two variants are entered into.

Common material is in the theory 'grice' whose listing may be found in appendix A.

SML

```
open_theory "rbjmisc";
force_new_theory "grice";
```

3 The Problem

Grice uses several lists in his presentation. To simplify unambiguous reference I have prefixed some of the numbering schemes with capital letters.

Our starting point is a list of eight *inclinations* supplied by Grice and summarised here:

I1 That individual constants be admitted.

I2 Note that names are sometimes "vacuous"

I3 Thence that a constant might lack a designatum.

I4 That excluded middle and bi-valence nevertheless remain unqualified.

I5 That a claim about (predication to) a non-designating constant be false and its negation true.

I6 That no unusual constraints on "U.I." [1] and "E.G." [2] be introduced.

I7 That the law of identity (in the form $\forall x \bullet x = x$) be a theorem[3].

I8 That derivability implies entailment.[4]

In relation to these Grice points out two *difficulties* as follows:

[1] Universal Instantiation

[2] Existential Generalisation

[3] Which seems to suggest that identity is not a predicate.

[4] i.e. that the system be sound!

(a) I2, I3 and I7 between them seem *prima facie* to enable proofs of the existence of non-existent entities (e.g. "Pegasus exists").

(b) I5 and I6 would if gratified permit us to infer that there exists something which does not fly from the premise that Pegasus does not exist.

He then lists five possible *resolutions* the difficulties:

R1 To resist I3 and insist that constants have a designatum.

R2 Resisting I4 and I5, insist that predications to non designating terms and their negations lack a truth value.

R3 Resist I1 and do without individual constants.

R4 Resist I2 by insisting that all constants have a designatum, which might perhaps "be" without existing.

R5 Resist I6 by adding requirements on existential generalisation that the relevant constant be shown to designate.

R6 Resist I8, allowing that the deductive system be not strictly sound, but only so subject to the condition (or "marginal" assumption) that all names have bearers.

Grice does not discuss any of these alternatives, but instead proposes to attempt to square the circle by devising a system which satisfies all these (apparently) incompatible inclinations.

4 System Q: Objectives

Grice now proposes a first order predicate calculus meeting two particular objectives, spelt out in five points of further detail.

(i) That a sentence such as "Pegasus does not fly" be capable of rendition in two distinct ways one of which will be true and the other false in the case that Pegasus does not exist. These correspond to disambiguations of the logical structure of the sentence either as predicating non-flying or denying a predication as flying.

(ii) That in either case the inference from "Pegasus does not fly" to "there is something which does not fly" is to be admitted.

O1 U.I. and E.G. are acceptable without special side conditions.

O2 Some sentences involving non-denoting constants will be true and provable.

O3 It will be formally decidable whether a sentence depends for its truth on whether some constant designates.

O4 It will be possible to find in Q representation of sentences such as "Pegasus exists".

O5 There will be an extension of Q in which identity is represented.[5]

[5] No discussion here of whether identity is a predicate.

5 Classical Logic using HOL

Grice tries to make his system as close as possible to "classical logic", basing his system on a first order predicate calculus presented as a natural deduction system based on a textbook by Mates [8]. I hope that the pertinent sense of "classical" here is *two-valued*.

In this presentation we work with a "classical" higher order logic (based on Church's formulation of the Simple Theory of Types[1]) in the form of a sequent calculus. In this section we replicate in this system theorems which show the closeness of the relationship between this classical logic and the one from which Grice departs.

Grice discusses the problem of the scope of names before getting into the details of his classical logic, but since we here adopt a different manner of resolving this problem which depends upon special features of our "classical logic", the discussion of scope belongs here.

5.1 Scope

A part of the novelty and difficulty in system Q lies in its special provisions for controlling the "scope" of names.

The use of the word scope in this context is distinct from its more common usage (at least in mathematical logic and computer science) in which the (or a) scope of a name (variable or constant) is a syntactic region within which it may be used with the same sense (as opposed to uses of the same name for entirely distinct purposes as may appear in different scopes of the name).

In this case the relevant ambiguity of scope is not a region of significance of the name, but the extent of some predicate applied to the name. Thus in the example used by Grice, there is an ambiguity in the sentence "Pegasus does not fly" about the logical structure of the sentence which leaves doubt about what predicate is being applied to Pegasus. We may regard the sentences as predicating "x does not fly" of Pegasus, or we may be denying that the predicate "flies" applies to Pegasus. The question is whether the negation is part of the predicate or is applied to the result of the predication.

In System Q Grice provides a predicate logic in which numerical subscripts may be used to disambiguate the predications which are taking place. Unfortunately, this notation, like the dot notation in Principia Mathematica is not as readable as the more usual use of brackets for disambiguating scope. Furthermore, this aspect of the syntax would be hard to replicate by the methods we propose to adopt, which are intended to permit a lightweight analysis of semantic issues and their connection with the validity of sentences and the soundness of derivations.

In our analysis we will therefore provide for the desired control over predication by the use of Church's lambda notation. We expect that our syntax will be readily related to that of System Q, but since the system we formalise is at least syntactically distinct we will call it system C[6].

System C is obtained by conservative extension to HOL, which is a version of higher order logic derived directly from Church's Simple Theory of Types [1]. Church's STT is distinctive for its very slender logical core, achieved by building the logical system from the simply typed lambda calculus. In this logical system boolean valued functions serve for predication, and the ideas of Grice on vacuous names are realised principally by defining a different kind of predication. To minimise confusion, the word predicate will only be used when Grice's notion of predication is intended, otherwise propositional or boolean valued function will be used.

The use of lambda notation for making explicit the predicate to be applied may be illustrated simply

[6] This in homage to Rudolf Carnap, Q having been chosen by Grice in homage to Quine, and G having been chosen by Myro[3, 9] in homage to Grice.

using Grice's Pegasus example. In using this method we must distinguish two kinds of predication. The kind whose scope we are now discussing, which is the only kind of predication in System Q, and the kind of predication which we have in HOL, consisting in the application of a propositional function to some argument. Henceforth these two will be distinguished by using the term "predication" only for the former, by making this kind of predication explicit in the concrete syntax as a copula, and by using the phrase "application of a propositional function" for the second kind of predication. The use of a copula not only visually distinguishes Griceian predication, but also allows us to control its semantics through the definition of the copula.

The copula will be the word "is", and since we will be discussion more than one variant of Grice's system, the copulas will be distinguished by subscripts giving the name of the system to which they belong. In System C, then, the copula is "is_c", so the application of the predicate "*Flier*" to the term "*Pegasus*" will be written "$Pegasus\ is_c\ Flier$".

The two variants of "Pegasus does not fly" are then expressed as:

"$\neg\ (Pegasus\ is_c\ Flier)$ "

and

"$Pegasus\ is_c\ (\lambda x \bullet\ \neg\ Flier\ x)$".

In the first case the predicate applied to Pegasus is $\ulcorner Flier \urcorner$, in the second case it is $\ulcorner \lambda x \bullet \neg Flier\ x \urcorner$ (which is a negation of Flier, "not a flier").

This technique makes it possible to examine the semantic aspects of System Q using formal machine assisted deductive reasoning, while avoiding the special intricacies of formalising Grice's special syntax. Whether this pays off in terms of perspicuity and penetration in the analysis remains to be seen.

We will therefore offer no further discussion of Grices notational devices for scoping at this point, but may consider later how much may have been lost by this approach.

At this stage I am looking at two alternatives for the underpining semantics, and the following two section work through these two alternatives. In each case, we are devising a "shallow embedding" of something close to Grice's System Q in ProofPower HOL. These embedded aproximations to System Q are called System C (in homage to Rudolf Carnap) and System S (a nod to Speranza).

5.2 Semantic Domains

Grice goes on from his discussions of scope to the presentation of the remaining aspects of the syntax, notably of the various aspects of the deductive system.

We will track this discussion of inference in a semantic way, demonstrating the semantic principles which ensure the soundness of the deductive system presented in a manner which makes reasonably clear the relationship between the two. This amounts to a semantic embedding into HOL of a system which is intended to be semantically parallel to system Q, though not syntactically quite the same.

To do this we must first provide some basic definitions on which the semantics is based, and before doing that I must explain part of the method of semantic embedding which might seem to, but does not in fact, prejudice one of the principle issues at stake. That is that the embedding consists in a development of a semantics which is purely denotational, i.e. in which every expression has a denotatum. These denotation need not be what one would naturally suppose the reference of an expression to be, they may be purely abstract surrogates which suffice to describe the truth conditions of the language. So the way we model a language in which term expressions may not denote, is by taking the range of things which they might possibly denote, and adding one or more extra items which are values which they are give to indicate that they have no denotation.

5.2.1 Interpretations

Grice's definition of an interpretation of Q (VIII A of [4]) is ambiguous in what it says about the domain of an interpretation. The domain is a set of correlates, some of which may be unit sets the element of which is a designatum. Grice tells us that there need not be any such unit sets, but does not say explicitly that there may be no other correlates. He does tell us that he has in mind that there will be exactly one non-designating correlate, the empty set, but he does not place this as a requirement, so the significance of his having this in mind is unclear.

There are three possibilities concerning the number of non-designating correlates in an interpretation, that there are none, exactly one, or more than one. In relation to these there are a number of possible positions in relation to the definition of an interpretation, the most obvious are:

1. no constraint
2. there must be at least one
3. there must be exactly one

Of course there may be more, but I propose to discuss just these three.

From a literal reading of Grice in this section I would take the first. However, from the fact that he explicitly allows there to be no designata but does not explicitly say this for non-designating correlates, we might infer that he intends but does not state that there will be at least one non-designating correlate. Alternatively we might take the case he had in mind more presciptively and work with item 3.

These are significant semantically, of course.

Allowing interpretations with no designata will create problems with descriptions if these were to be introduced into Q, for one would then have no correlate available for a non-designating descriptions. It also affects a matter which Grice later discusses, which is whether certain existential claims are valid, notably the claim that there is something which does not exist. This (or something similar " $\exists x_{4}.\ \neg_{3}\ F_{1}\ x_{2}''$", VIII B p137) Grice later claims to be valid, and this is evidence against interpretation 1 in favour of 2 or 3.

Item 2, though the one perhaps preferred by Grice, has the disadvantage that all non-designating terms have the same correlate, and will therefore be equal in a sense of equality which satisfies the universal law of reflexivity of identity. Do we want Pegasus and Sherlock Holmes to be identical?

Nevertheless, in our shallow embedding of Q we will adopt 2, for, in addition to its possible preferment by Grice, and its confirming Grice's belief in the validity of the cited existentials, it allows a simpler and more transparent shallow embedding than either of the others. *[fuller explanation to be supplied]*

First let us give a name for the domain of discourse of an interpretation of the language Q. We will suppose this to be a "type" in HOL, but will leave open the type by using a type variable. To serve as denotions for terms in system G_{HP} we use a disjoint union of two types, the first of which is the type of those things which do exist (candidate designata) and the second is a type of surrogates ("correllates" in Grice) for those things which do not exist. We use V for Value.

5.3 Descriptions

When Grice comes to discuss descriptions he has in mind a distinction between descriptions and names which would impact upon the kind of interpretation necessary for a semantic embedding. This in turn would impact the detailed definition of the copula.

To make apparent the differences which would arise in treating descriptions as Grice intends we provide two treatments of those aspects of the logical system which depend upon the details of the semantics. These will be called System C and System S.

Certain aspects of our presentation, particularly those which simply illustrate the correspondence between the classical propositional logic in System Q and that in ProofPower-HOL, are common to System C and System S and are therefore treated first.

5.4 Propositional Logic

The following material is common to Systems C and S and shows how these systems relate to the propositional logic in System Q. Since we are working with a semantic embedding rather than a syntactic meta-theory we cannot prove results about derivations and instead present proofs about the semantics which show that the required inference rules would be sound.

To do this we adopt a notation for expressing entailment in C in a manner parallel to the presentation of the natural deduction system of Q. This is done by defining the infix relationship $(al \models c)$ holding between a list of assumptions (or premises) al and a single conclusion c when the conclusion is entailed by the premises.

SML

```
│declare_infix(4, "⊨");
```

HOL Constant

```
│ $⊨ : BOOL LIST → BOOL → BOOL
├──────────────────────────────
│ ∀al c• (al ⊨ c) ⇔ (∀_L al) ⇒ c
```

In the following theorems which capture the content of Grices rules for Q as theorems of C, the variables ϕ, ψ, ζ... range over formulae (boolean terms) and Γ, Δ, Θ... over lists (finite sequences) of formulae. P ranges over propositional functions, values whose types are instances of ⌜:'a → BOOL⌝ (in which $'a$ is a type variable). The notation “$[\phi; \psi; ...]$” is an explicit list formation (“$[\phi]$” being a single element list) and the symbol “$\frown$” is the concatenation operator over lists.

In general the rules of Q can be formulated as theorems of HOL because in a higher order logic, and can quantify over predicates. This means that where substitution is involved in a rule, this is captured as a theorem by the use of a variable as a predicate applied to a variable as an argument. When such a theorem is instantiated using a lambda expression as the predicate, the substitution into the expression will be achieved by beta-reduction.

$\boldsymbol{ASS} = \quad \vdash \forall\ \phi \bullet\ [\phi] \models \phi$

$\boldsymbol{RAA} = \quad \vdash \forall \phi\ \psi\ \zeta\ \mathit{\Gamma} \bullet\ ([\phi] \frown \mathit{\Gamma} \models \psi \land \neg\psi) \Rightarrow (\mathit{\Gamma} \models \neg\phi)$

$\boldsymbol{DN} = \quad \vdash \forall\ \phi \bullet\ [\neg\ \neg\ \phi] \models \phi$

$\land\boldsymbol{I} = \quad \vdash \forall\ \phi\ \psi \bullet\ [\phi; \psi] \models \phi \land \psi$

$\land\boldsymbol{E} = \quad \vdash \forall\ \phi\ \psi \bullet\ ([\phi \land \psi] \models \phi) \land ([\phi \land \psi] \models \psi)$

$\lor\boldsymbol{I} = \quad \vdash \forall\ \phi\ \psi \bullet\ ([\phi] \models \phi \lor \psi) \land ([\phi] \models \psi \lor \phi)$

$\lor\boldsymbol{E} = \quad \vdash \forall\ \phi\ \psi\ \zeta\ \mathit{\Gamma}\ \mathit{\Delta}\ \mathit{\Theta} \bullet\ ([\phi] \frown \mathit{\Gamma} \models \zeta) \land ([\psi] \frown \mathit{\Delta} \models \zeta) \land (\mathit{\Theta} \models \phi \lor \psi)$
$\quad\quad \Rightarrow (\mathit{\Gamma} \frown \mathit{\Delta} \frown \mathit{\Theta} \models \zeta)$

$\boldsymbol{CP} = \quad \vdash \forall\ \phi\ \psi\ \mathit{\Gamma} \bullet\ ([\phi] \frown \mathit{\Gamma} \models \psi) \Rightarrow (\mathit{\Gamma} \models \phi \Rightarrow \psi)$

$\boldsymbol{MPP} = \quad \vdash \forall\ \phi\ \psi \bullet\ [\phi \Rightarrow \psi; \phi] \models \psi$

5.5 Quantifier Inference-Rules

We now show the correspondence between the quantification rule in System Q with the standard quantifiers in HOL. Because of this correspondence we can use the standard HOL quantifiers in System C and get a good correspondence with System Q in this area. When we come to System S it will be necessary to redefine the quantifiers to obtain those qualifications on $\forall E$ and $\exists I$ which are necessary to cope with Grice's ideas about failing descriptions.

$\forall\boldsymbol{I} = \quad \vdash \forall\ \mathit{\Gamma}\ P \bullet\ (\forall\ x \bullet\ \mathit{\Gamma} \models P\ x) \Rightarrow (\mathit{\Gamma} \models (\forall\ x \bullet\ P\ x))$

$\forall\boldsymbol{E} = \quad \vdash \forall\ \mathit{\Gamma}\ P\ c \bullet\ (\mathit{\Gamma} \models (\forall\ x \bullet\ P\ x)) \Rightarrow (\mathit{\Gamma} \models P\ c)$

$\exists\boldsymbol{I} = \quad \vdash \forall\ \mathit{\Gamma}\ P\ x \bullet\ (\mathit{\Gamma} \models P\ x) \Rightarrow (\mathit{\Gamma} \models (\exists\ x \bullet\ P\ x))$

$\exists\boldsymbol{E} = \quad \vdash \forall\ P\ Q \bullet\ (\forall\ x \bullet\ [P\ x] \models Q) \Rightarrow ([\exists\ x \bullet\ P\ x] \models Q)$

5.6 P Committal

The notion of E-committal is defined syntactically by Grice, and tells us which formulae in System Q are true only if some name has a designatum, i.e. to the existence of which named individuals our assertion of the formula would commit us.

In HOL we can formalise this idea without resort to reasoning about syntax, furthermore, the main detail in the definition is independent of whether we are in System C or System S, and is therefore provided here. To achieve this we parameterise the definition of E-committal so that it can be specialised to the different ways of asserting existence in Systems C and S, the parameterised version being called *P_committal*.

The formal criteria for *E_committal* then becomes provability in HOL (which is not in general decidable).

In Grice *E_committal* is a relationship, here it is a parameterised property of propositional functions (the parameter being the relevant notion of existence).

SML

```
declare_infix(100, "committal");
```

HOL Constant

$\$\boldsymbol{committal} : ('a \to BOOL) \to ('a \to BOOL) \to BOOL$

$\forall P\ \phi \bullet\ P\ committal\ \phi \Leftrightarrow \forall \alpha \bullet\ \phi\ \alpha \Rightarrow P\ \alpha$

The following theorems correspond to clauses in Grice's definition of *E_committal*:

$$\begin{array}{ll}
\boldsymbol{P\_2} = & \vdash \forall\ P\ \phi\bullet\ P\ committal\ \phi \\
 & \qquad \Rightarrow P\ committal\ (\lambda\alpha\bullet\ \neg\ \neg\ \phi\ \alpha) \\
\boldsymbol{P\_3} = & \vdash \forall\ P\ \phi\ \psi\bullet\ P\ committal\ \phi\ \vee\ P\ committal\ \psi \\
 & \qquad \Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \phi\ \alpha\ \wedge\ \psi\ \alpha) \\
\boldsymbol{P\_4} = & \vdash \forall\ P\ \phi\ \psi\bullet\ P\ committal\ \phi\ \wedge\ P\ committal\ \psi \\
 & \qquad \Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \phi\ \alpha\ \vee\ \psi\ \alpha) \\
\boldsymbol{P\_5} = & \vdash \forall\ P\ \phi\ \psi\bullet\ P\ committal\ (\lambda\ \alpha\bullet\ \neg\ \phi\ \alpha)\ \wedge\ P\ committal\ \psi \\
 & \qquad \Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \phi\ \alpha\ \Rightarrow\ \psi\ \alpha) \\
\boldsymbol{P\_6}\forall = & \vdash \forall\ P\ \phi\bullet\ (\exists\ \beta\bullet\ P\ committal\ (\phi\ \beta)) \\
 & \qquad \Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \forall\ \beta\bullet\ \phi\ \beta\ \alpha)
\end{array}$$

I have a problem proving the existential element of (6), so this clause is in doubt. I think Grice should be asking for "$\forall\beta\bullet\ \psi(\beta/\omega)$" to be *E_committal* for α (in system Q), which corresponds in systems C and S to the theorem:

$$\begin{array}{ll}
\boldsymbol{P\_6\exists b} = & \vdash \forall\ P\ \phi\bullet\ (\forall\ \beta\bullet\ P\ committal\ (\phi\ \beta)) \\
 & \qquad \Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \exists\ \beta\bullet\ \phi\ \beta\ \alpha)
\end{array}$$

(with P suitably instantiated).

6 System C

A new theory is needed which I will call "griceC" which is created here:

SML

```
open_theory "grice";
force_new_theory "griceC";
set_pc "rbjmisc";
```

Grice introduces a special notion of interpretation in terms of which to give the semantics of Q. An interpretation has a domain which is a set of correlates, some of which are unit sets, in which case the member of that unit set is called a designatum. For uniformity of type we assume that all correlates are sets of sets.

We allow the domain of an interpretation to be any type and for some element of that type to be identified as THE non-designating element.

HOL Constant

```
⊥: 'a
────────────────────────────
T
```

We can then use inequality with ⊥ as the property of being a designatum. The structure of these interpretations is isomorphic to that of Grice's preferred notion of interpretation, even though we do not use unit sets for correlates of designating constants.

We may use HOL constants of appropriate type for Q constants and predicates, and HOL variables for Q variables, and the consequence should be that the theorems provable in HOL should be just those which are valid in Q.

This is done by using T and F instead of Corr(1) and Corr(0) respectively.

Now we define our special kind of predication. This is done by introducing ‘is’ subscripted with ‘c’ as a copula.

SML

```
declare_infix (400, "is_c");
```

HOL Constant

```
$is_c: 'a → ('a → BOOL) → BOOL

∀ p t• t is_c p ⇔ if t = ⊥ then F else p t
```

The definition says, if the term denotes then apply the predicate to the value denoted, otherwise the result of the predication is F (false).

Note that this will not work for relations, we would have to define a separate similar predicator for each arity of relation in use.

Here is the copula for applying a 2-ary relation.

SML

```
declare_infix (400, "is_c2");
```

HOL Constant

```
$is_c2: ('a × 'b) → ('a × 'b → BOOL) → BOOL

∀ p t• t is_c2 p ⇔ if Fst t = ⊥ ∨ Snd t = ⊥ then F else p t
```

Now let’s introduce Pegasus. The “definition” just says that Pegasus is the non-denoting value.

HOL Constant

```
Pegasus : 'a

Pegasus = ⊥
```

And the predicate *Flier*. Note that the predicates apply only to real desigata under the normal HOL application, its only under our special predication that they become applicatble to possibly non denoting values.

HOL Constant

```
Flier : 'a → BOOL

T
```

We can now prove in HOL the elementary facts we know about Pegasus flying, yielding the following theorems:

Pegasus_lemma_1 =

$\vdash \mathit{Pegasus}\ \mathit{is}_c\ \mathit{Flier} \Leftrightarrow F$

Pegasus_lemma_2 =

$\vdash \mathit{Pegasus}\ \mathit{is}_c\ (\lambda\ x\bullet\ \neg\ \mathit{Flier}\ x) \Leftrightarrow F$

Pegasus_lemma_3 =

$\vdash \neg\ \mathit{Pegasus}\ \mathit{is}_c\ \mathit{Flier}$

Pegasus_lemma_4 =

$\vdash \neg\ \mathit{Pegasus}\ \mathit{is}_c\ (\lambda\ x\bullet\ \neg\ \mathit{Flier}\ x)$

In case you may doubt the neutrality of our use of lambda expressions, we show here that they are inessential to this approach, and can be dispensed with in favour of explict definitions of the predicates required. Thus we may instead define a predicate which is the denial of *Flier*:

HOL Constant

$\boldsymbol{NonFlier} : {}'a \to \mathit{BOOL}$

$\forall x\bullet\ \mathit{NonFlier}\ x \Leftrightarrow \neg\ \mathit{Flier}\ x$

We can then prove the slightly reworded:

Pegasus_lemma_5 = $\vdash \mathit{Pegasus}\ \mathit{is}_c\ \mathit{NonFlier} \Leftrightarrow F$

the proof of which need not mention the predicate, since it suffices to know that the subject does not exist.

This general claim can be expressed in HOL and proven. To make it read sensibly however, it is best for us to define the relevant notion of existence.

HOL Constant

$\boldsymbol{Existent} : {}'a \to \mathit{BOOL}$

$\forall x\bullet\ \mathit{Existent}\ x \Leftrightarrow T$

The definition looks a bit bizarre and works for just the same reason that we do not need to know the predicate to know that predication to a non-existent will be false. This predicate will always be true, unless it is applied to a term which fails to denote. So we satisfy another of Grice's desiderata, that the non-existence of Pegasus can be stated.

But first a more general result:

Existent_lemma_1 =

$\vdash \forall\ P\ x\bullet\ \neg\ x\ \mathit{is}_c\ \mathit{Existent} \Rightarrow \neg\ x\ \mathit{is}_c\ P$

Desideratum A8 is satisfied by the standard quantifiers in HOL.

$\exists$***_lemma_1*** =

$\vdash \forall\ P\bullet\ \mathit{Pegasus}\ \mathit{is}_c\ P \Rightarrow (\exists\ x\bullet\ x\ \mathit{is}_c\ P)$

$\forall$***_lemma_1*** =

$\vdash \forall\ P\bullet\ (\forall\ x\bullet\ x\ \mathit{is}_c\ P) \Rightarrow \mathit{Pegasus}\ \mathit{is}_c\ P$

6.1 Quantifier Inference-Rules

An example of the effect of substitutions obtained by use of these theorems in C intended to mimic the rules of Q is as follows.

Take ths simple case of $\forall E$, with the predicate $\lambda x \bullet\ \neg\ x\ is_c\ Flier$, which we seek to instantiate to Pegasus.

We can specialise $\forall E$ to this case like this:

SML

```
val sve = list_∀_elim [⌜[]:BOOL LIST⌝, ⌜λx• ¬ x is_c Flier⌝, ⌜Pegasus⌝] ∀E;
```

This gives the theorem:

```
val sve =   ⊢ ([] ⊨ (∀ x• (λ x• ¬ x is_c Flier) x))
            ⇒ ([] ⊨ (λ x• ¬ x is_c Flier) Pegasus) : THM
```

in which there are two occurrences of the lambda expression in the places previously occupied by the variable P.

The substitutions can then be effected by beta reduction which can be achieved by rewriting the theorem as follows:

SML

```
rewrite_rule [] sve;
```

Which yields a theorem justifying the required inference.

```
val it = ⊢ ([] ⊨ (∀ x• ¬ x is_c Flier)) ⇒ ([] ⊨ ¬ Pegasus is_c Flier) : THM
```

All the above rules simply explicate the standard aspects of the logic of Q and C and do not reflect the specific additional features which Grice requires in connection with vacuous names. They simply illustrate that the logical context for C includes as much standard logic as is provided in Q, and provide some tenuous indication that Grice's rules conform to standard logic as defined in Church's simple theory of types. The main risk for Grice is in the detail of his presentation, bearing in mind the additional complexities arising from his suffix notation, and the above exercise offers no assurance in that matter.

6.2 Existence

6.2.1 A. (E Committal)

We have already defined the predicate "Existent" which tells us whether a name designates.

The following theorem indicates that the propositional function captures the intent of Grice's "+exists" notation.

```
E1 =⊢ ∀φ α• α is_c φ ⇒ α is_c Existent
```

E Committal Grice's notion of E committal corresponds in System C to the previously defined generic notion of committal instantiated to the propositional function *Existent.*

The definition we have already given for *Existent* is similar in character and identical in effect to Grices suggestions for *+exists.*

6.2.2 B. Existentially Quantified Formulae

This has already been covered.

Grice wishes to distinguish himself from Meinong, but his system allows quantification over non-existents, and he can therefore only distinguish himself from Meinong by distancing himself also from Quine's criterion of ontological committment. This he could easily do by adopting Carnap's distinction between internal and external questions, and denying that any metaphysical significance attaches to something being counted in the range of quantification.

6.3 Identity

HOL already has identity and this identity is good for System C. In HOL identity is a curried propositional function of polymorphic type ⌜:'a → 'a → BOOL⌝. If applied using function application the universal law of reflexivity holds.

To get a "strict" version of equality (i.e. one which is false when either argument does not exist) it is necessary to uncurry equality so that its type becomes ⌜:('a × 'a) → BOOL⌝ and apply it using the system C copula for binary relations.

These two seem to correspond to Grice's intended two kinds of equality, insofar as the difference between the two is primarily that one does and the other does not allow that there may be true identities between non-denoting names. However this is achieved in system C by having two quite distinct identity relations only one of which is properly a predicate, and applied through our copula is_c. The other is a propositional function which must be applied without use of the copula (because the copula forces strictness, i.e. forces predication to undefined values to yield falsehoods).

Grice seems to think that the required distinction between equalities can be achieved by the use of his scoping subscripts, but I have not yet been convinced that he is correct in this. Whatever the scope of the predicate, predication to non-denoting names must yield falsehood, so I can't see how any use of subscripts will make "Pegasus = Pegasus" to be true.

To illustrate the effect of the built-in equality the following theorems suffice:

***Pegasus_eq_lemma*1** =

$\vdash Pegasus = Pegasus$

Note the lack of copula. This is of course an instance of the theorem:

eq_refl_thm =

$\vdash \forall\ x\bullet\ x = x$

To get a nice notation for "strong" equality we need to define it as an infix symbol, which requires that the copula be built into the definition. We could use the ≡ symbol, but this usually represents an equivalence relation which is weaker than identity, so we will subscript the = sign with c, suggesting that it is just the same thing applied through our copula.

SML

$declare\_infix\ (200,\ "=_c");$

HOL Constant

$\$=_{\boldsymbol{c}} : {}'a \to {}'a \to BOOL$

$\forall x\ \ y{:}'a\bullet\ (x =_c y) = (x{,}y)\ is_{c2}\ (Uncurry\ \$=)$

We can now prove:

$$\begin{array}{|l} \mathit{Pegasus\_eq\_lemma2}\ = \\ \qquad \vdash \neg\ \mathit{Pegasus} =_c \mathit{Pegasus} \end{array}$$

And must rest content with the qualified rule:

$$\begin{array}{|l} \boldsymbol{eq_c\_refl\_thm}\ = \\ \qquad \vdash \forall\ x\bullet\ x\ is_c\ \mathit{Existent} \Leftrightarrow x =_c x \end{array}$$

The "strictness" of this equality is expressed by the theorem (which would have made a better definition):

$$\begin{array}{|l} eq_c\_strict\_thm\ = \\ \qquad \vdash \forall\ x\ y\bullet\ x =_c y \Leftrightarrow x\ is_c\ \mathit{Existent} \wedge y\ is_c\ \mathit{Existent} \wedge x = y \end{array}$$

On reflection Grice's treatment by scoping is presented in the context of a second order definition of equality, and therefore for the weak equality does not define equality as a predicate, but as a second order formula. Its equivalence to the primitive equality in HOL can be illustrated by the following theorem:

$$\begin{array}{|l} \boldsymbol{snd\_order\_eq\_thm}\ = \\ \qquad \vdash \forall\ x\ y\bullet\ x = y \Leftrightarrow (\forall\ P\bullet\ x\ is_c\ P \Leftrightarrow y\ is_c\ P) \end{array}$$

This equivalence only holds because we adopted Grice's preferred (but not stipulated) domain of interpretation, in which there is exactly one non-designating correlate. If there was more than one non-designating correlate then Grice's second order definition would give a notion of identity under which all non-designating correlates are considered equal, which would not be the case for the primitive equality in HOL.

Even in that case a similar device would work as illuatrated by the following theorem:

$$\begin{array}{|l} \boldsymbol{snd\_order\_eq\_thm2}\ = \\ \qquad \vdash \forall\ x\ y\bullet\ x = y \Leftrightarrow (\forall\ P\bullet\ P\ x \Leftrightarrow P\ y) \end{array}$$

In which the griceian copula has been dropped so that the predication is no longer strict.

6.4 Names and Descriptions

I note here pro-tem just that the method adopted in System C for controlling the scope of predications, viz. the use either of lambda expressions or of naming complex predicates by definitions, enables scope to be controlled in predication to descriptions which are formed as terms. We can therefore follow Grice's system Q in introducing such descriptions.

SML

```
declare_binder "ι";
```

HOL Constant

$$\begin{array}{|l} \$\boldsymbol{\iota} : ('a \to BOOL) \to 'a \\ \hline \forall \phi\bullet\ (\$\iota\ \phi) = \epsilon\alpha\bullet\ (\alpha\ is_c\ \phi \wedge \forall\beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta = \alpha) \\ \qquad\qquad \vee\ (\neg\exists\gamma\bullet\ \gamma\ is_c\ \phi \wedge \forall\beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta = \gamma) \wedge \alpha = \bot \end{array}$$

$\boldsymbol{\iota\_thm1} = \vdash \forall\ \gamma\bullet\ \gamma\ is_c\ \phi \land (\forall\ \beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta = \gamma) \Rightarrow \$\iota\ \phi = \gamma$

$\boldsymbol{\iota\_thm2} = \vdash \neg\ (\exists\ \gamma\bullet\ \gamma\ is_c\ \phi \land (\forall\ \beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta = \gamma)) \Rightarrow \$\iota\ \phi = \bot$

There is a difficulty in adopting Grice's reticence to allow existential generalisation on the basis of such a predication. This cannot be supported in system C without changing the underlying semantics.

Grice has effectively split the non-denoting expressions into two, those which correlate with something in the domain of quantification and those which do not. Since our embedding requires all expressions to have a value, we would have to add another kind of entity into the domain of interpretation and define quantifiers which operate over expressions which have a correlate but not over the values of non-denoting descriptions.

This would effect most of the embedding, the whole would have to be reworked (in a routine way). We investigate this possibility in the next section.

7 System S

System S contains two minor adjustments to System C which are concerned with the treatment of non-designating names and failing descriptions. In relation to non-designating names, the semantics in System S now allows that they have distinct correlates. In addition a new subdomain is introduced which provides at least one (since we can always prove that there is a failing description) surrogate for a failing description. The primary reason for recognising, in the semantics, denotations for failing descriptions, is so that we can arrange as Grice seems to require, that they do not fall into the range of the System S quantifiers. This in turn results in the qualification of the rules EG and UI.

A new theory is needed which I will call "griceS" which is created here:

SML

```
open_theory "grice";
force_new_theory "griceS";
set_pc "rbjmisc";
```

We liberalise the notion of interpretation from system C in the following two ways. Firstly we allow that there may me more than one non-denoting correllate. and hence then there may be semantic distinctions between non-denoting names, overcoming one obstacle to substantial discussions about fictional entities. Secondly we introduce values to stand for non-referring descriptions. These are not intended to be values and will be excluded from the range of quantification, with the effect that EG and UI will not work with failing descriptions.

This will be done using a "disjoint union" the right disjunct of which will be surrogates for failing descriptions, and the left will be correlates for names of which there will be at least one which is not a denotatum. The "normal" quantifiers for system S will quantify only over the correlates.

To effect this we introduce the property of being a denotatum which is left open except to the extent that there must be at least one correlate which is not a denotatum.

HOL Constant

$\boldsymbol{denotatum}: {'a} \to BOOL$

$\exists x\bullet\ \neg\ denotatum\ x$

HOL Constant

```
│ ⊥_corr: 'a
├──────────────────────────────
│ ¬ denotatum ⊥_corr
```

The type of exressions in System S is given by the following type abbreviation:

SML

```
│declare_type_abbrev("SC", [], ⌜:'a + 'b⌝);
```

HOL Constant

```
│ ⊥: SC
├──────────────────────────────
│ ⊥ = InL ⊥_corr
```

It will be useful to have tests and projections for values of type ⌜:SC⌝.

This propositional function tests whether an SC is a correlate:

HOL Constant

```
│ IsCorr: SC → BOOL
├──────────────────────────────
│ ∀v:SC• IsCorr v ⇔ IsL v
```

This function extracts the correlate (of type ⌜:'a⌝) from a value of type ⌜:SC⌝.

HOL Constant

```
│ Corr: SC → 'a
├──────────────────────────────
│ ∀v:SC• Corr v = OutL v
```

This propositional function tells us whether a value of type ⌜:SC⌝ is a denotatum.

HOL Constant

```
│ IsDen: SC → BOOL
├──────────────────────────────
│ ∀v:SC• IsDen v ⇔ IsCorr v ∧ denotatum (Corr v)
```

There are now difficulties in the treatment of constants and variables, since a HOL constant or variable constrained only by the type of the elements of an interpretation will not be known to denote (which is OK), but will not even be known to correlate, and hence may not even be in the range of quantification. Such results as can be obtained using unconstrained variables will nevertheless be valid, and will be generalisable.

Now we define our special kind of predication (the copula for System S).

SML

```
│declare_infix (400, "is_s");
```

HOL Constant

```
│ $is_s: SC → ('a → BOOL) → BOOL
├──────────────────────────────────────
│ ∀ p t• t is_s p ⇔ if IsDen t then p (Corr t) else F
```

The definition says, if the term denotes then apply the predicate to the value denoted, otherwise the result of the predication is F (false).

Note that this will not work for relations, we would have to define a separate similar predicator for each arity of relation in use.

Here is the copula for applying a 2-ary relation.

SML

```
│declare_infix (400, "is_s2");
```

HOL Constant

```
│ $is_s2: (SC × SC) → ('a × 'a → BOOL) → BOOL
├──────────────────────────────────────
│ ∀ p t• t is_s2 p ⇔       if IsDen (Fst t) ∧ IsDen (Snd t)
│                    then p (Corr (Fst t), Corr (Snd t))
│                    else F
```

Now let's introduce Pegasus. The "definition" just says that Pegasus is a non-denoting correlate. We know that there must be such a thing (though we don't know that there are any denoting correlates).

HOL Constant

```
│ Pegasus : SC
├──────────────────────────────────────
│ IsCorr Pegasus ∧ ¬ IsDen Pegasus
```

And the predicate *Flier*. Note that the predicates apply only to real desigata under the normal HOL application, its only under our special predication that they become applicatble to possibly non denoting values.

HOL Constant

```
│ Flier : 'a → BOOL
├──────────────────────────────────────
│ T
```

We can now prove in HOL the elementary facts we know about Pegasus flying, yielding the following theorems:

$\boldsymbol{Pegasus\_lemma\_1}$ =

$\qquad \vdash Pegasus\ is_s\ Flier \Leftrightarrow F$

$\boldsymbol{Pegasus\_lemma\_2}$ =

$\qquad \vdash Pegasus\ is_s\ (\lambda\ x\bullet\ \neg\ Flier\ x) \Leftrightarrow F$

$\boldsymbol{Pegasus\_lemma\_3}$ =

$\qquad \vdash \neg\ Pegasus\ is_s\ Flier$

$\boldsymbol{Pegasus\_lemma\_4}$ =

$\qquad \vdash \neg\ Pegasus\ is_s\ (\lambda\ x\bullet\ \neg\ Flier\ x)$

In case you may doubt the neutrality of our use of lambda expressions, we show here that they are inessential to this approach, and can be dispensed with in favour of explict definitions of the predicates required. Thus we may instead define a predicate which is the denial of *Flier*:

HOL Constant

$\boldsymbol{NonFlier} : {}'a \to BOOL$

$\forall x\bullet\ NonFlier\ x \Leftrightarrow \neg\ Flier\ x$

We can then prove the slightly reworded:

$\boldsymbol{Pegasus\_lemma\_5} = \vdash Pegasus\ is_s\ NonFlier \Leftrightarrow F$

the proof of which need not mention the predicate, since it suffices to know that the subject does not exist.

This general claim can be expressed in HOL and proven. To make it read sensibly however, it is best for us to define the relevant notion of existence.

HOL Constant

$\boldsymbol{Existent} : {}'a \to BOOL$

$\forall x\bullet\ Existent\ x \Leftrightarrow T$

The definition looks a bit bizarre and works for just the same reason that we do not need to know the predicate to know that predication to a non-existent will be false. This predicate will always be true, unless it is applied to a term which fails to denote. So we satisfy another of Grice's desiderata, that the non-existence of Pegasus can be stated.

But first a more general result:

$\boldsymbol{Existent\_lemma\_1}$ =

$\qquad \vdash \forall\ P\ x\bullet\ \neg\ x\ is_s\ Existent \Rightarrow \neg\ x\ is_s\ P$

Desideratum A8 is satisfied by the standard quantifiers in HOL.

$\exists\boldsymbol{\_lemma\_1}$ =

$\qquad \vdash \forall\ P\bullet\ Pegasus\ is_s\ P \Rightarrow (\exists\ x\bullet\ x\ is_s\ P)$

$\forall\boldsymbol{\_lemma\_1}$ =

$\qquad \vdash \forall\ P\bullet\ (\forall\ x\bullet\ x\ is_s\ P) \Rightarrow Pegasus\ is_s\ P$

However, the HOL native quantifiers are unsatisfactory since they quantify over the values introduced into the domain of the interpretation as correlates for failing descriptions.

This is shown by the following theorem:

$$\exists\_\boldsymbol{not\_Corr\_thm} =$$
$$\vdash \exists\ x\bullet\ \neg\ IsCorr\ x$$

7.1 Quantifier Definitions

We now define quantifiers which quantify over correlates only, not over the values of failing descriptions.

Quantifiers are propositional functions over propositional functions, and are usually syntactically "binders", so that they are expected to be applied to a lambda abstraction and the lambda in that expression is to be omitted. We use the usual quantifier symbols subscripted with s to distinguish them from the standard quantifiers.

SML

declare_binder "$\forall_s$";
declare_binder "$\exists_s$";

The domain of interpretation now includes correlates and some other things (for failing descriptions). The things for failing descriptions are to be excluded from the range of the quantifiers (they really really don't exist) so quantifiers range only over correlates. The dollar sign locally supresses the binder status for the purpose of giving the definition.

HOL Constant

$$\$\forall_{\boldsymbol{s}} : (SC \to BOOL) \to BOOL$$

$$\forall f\bullet\ \$\forall_s\ f \Leftrightarrow \forall x\bullet\ IsCorr\ x \Rightarrow f\ x$$

HOL Constant

$$\$\exists_{\boldsymbol{s}} : (SC \to BOOL) \to BOOL$$

$$\forall f\bullet\ \$\exists_s\ f \Leftrightarrow \exists x\bullet\ IsCorr\ x \land f\ x$$

To express the rules for these new quantifiers we define a new notion of existence. This cannot be a bona-fide predicate, since such a predicate would of necessity be false of the correlates which are not designata (if applied through the copula), so it must be thought of as a propositional function to be applied without the copula.

HOL Constant

$$\boldsymbol{Existent_s} : SC \to BOOL$$

$$\forall x\bullet\ Existent_s\ x \Leftrightarrow \exists_s\ y\bullet\ x = y$$

7.2 Quantifier Inference-Rules

The propositional rules are common to Systems C and S and are shown above in Section 5.4.

We have new quantifiers in System S, and the quantifier rules will be modified accordingly.

$$\begin{array}{ll}
\forall \boldsymbol{I_s} = & \vdash \forall\ \mathit{\Gamma}\ P\bullet\ (\forall\ x\bullet\ \mathit{\Gamma} \models \mathit{Existent}_s\ x\ \wedge\ P\ x) \Rightarrow (\mathit{\Gamma} \models (\forall_s\ x\bullet\ P\ x)) \\
\forall \boldsymbol{E_s} = & \vdash \forall\ \mathit{\Gamma}\ P\ c\bullet\ (\mathit{\Gamma} \models \mathit{Existent}_s\ c\ \wedge\ (\forall_s\ x\bullet\ P\ x)) \Rightarrow (\mathit{\Gamma} \models P\ c) \\
\exists \boldsymbol{I_s} = & \vdash \forall\ \mathit{\Gamma}\ P\ x\bullet\ (\mathit{\Gamma} \models \mathit{Existent}_s\ x\ \wedge\ P\ x) \Rightarrow (\mathit{\Gamma} \models (\exists_s\ x\bullet\ P\ x)) \\
\exists \boldsymbol{E_s} = & \vdash \forall\ P\ Q\ c\bullet\ (\forall\ x\bullet\ [\mathit{Existent}_s\ x\ \wedge\ P\ x] \models Q) \Rightarrow ([\exists_s\ x\bullet\ P\ x] \models Q)
\end{array}$$

An example of the effect of substitutions obtained by use of these theorems in S intended to mimic the rules of Q is as follows.

Take ths simple case of $\forall E_s$, with the predicate $\lambda x \bullet\ \neg\ x\ \mathit{is}_s\ \mathit{Flier}$which we seek to instantiate to Pegasus.

We can specialise $\forall E_s$ to this case like this:

SML

$$\mathit{val\ sves} = \mathit{list\_\forall\_elim}\ [\ulcorner[]{:}\mathit{BOOL\ LIST}\urcorner,\ \ulcorner\lambda x{:}\ {'a}\ +\ {'b}\bullet\ \neg\ x\ \mathit{is}_s\ \mathit{Flier}\urcorner,\ \ulcorner\mathit{Pegasus}\urcorner]\ \forall E_s;$$

This gives the theorem:

$$\begin{array}{l}
\mathit{val\ sves} = \vdash ([] \models \mathit{Existent}_s\ \mathit{Pegasus}\ \wedge\ (\forall_s\ x\bullet\ (\lambda\ x\bullet\ \neg\ x\ \mathit{is}_s\ \mathit{Flier})\ x)) \\
\qquad\qquad\qquad \Rightarrow ([] \models (\lambda\ x\bullet\ \neg\ x\ \mathit{is}_s\ \mathit{Flier})\ \mathit{Pegasus}) : \mathit{THM}
\end{array}$$

in which there are two occurrences of the lambda expression in the places previously occupied by the variable P.

The substitutions can then be effected by beta reduction which can be achieved by rewriting the theorem as follows:

SML

$$\mathit{rewrite\_rule}\ []\ \mathit{sves};$$

Which yields a theorem justifying the required inference.

$$\begin{array}{l}
\mathit{val\ it} = \quad \vdash ([] \models \mathit{Existent}_s\ \mathit{Pegasus}\ \wedge\ (\forall_s\ x\bullet\ \neg\ x\ \mathit{is}_s\ \mathit{Flier})) \\
\qquad\qquad\qquad \Rightarrow ([] \models \neg\ \mathit{Pegasus}\ \mathit{is}_s\ \mathit{Flier}) : \mathit{THM}
\end{array}$$

7.3 Existence

7.3.1 A. (E Committal)

We have already defined the propositional function "Existent" which tells us whether a name designates.

The following theorem indicates that the propositional function captures the intent of Grice's "+exists" notation.

$$\boldsymbol{E1} = \vdash \forall\ \phi\ \alpha\bullet\ \alpha\ \mathit{is}_s\ \phi \Rightarrow \mathit{Existent}\ \alpha$$

Once again, Grice's notion of *E_committal* is obtainable for System S by instantiation of our generic notion of *committal* by the propositional function *Existent*.

The definition we have already given for *Existent* is similar in character and identical in effect to Grices suggestions for *+exists*.

7.3.2 B. Existentially Quantified Formulae

This has already been covered.

Grice wishes to distinguish himself from Meinong, but his system allows quantification over non-existents, and he can therefore only distinguish himself from Meinong by distancing himself also from Quine's criteria of ontological committment. This he could easily do by adopting Carnap's distinction between internal and external questions, and denying that any metaphysical significance attaches to something being counted in the range of quantification.

7.4 Identity

HOL already has identity and this identity is good for System C. It HOL it is a curried propositional function of polymorphic type ⌜:'a → 'a → BOOL⌝. If applied using function application the universal law of reflexivity holds.

To get a "strict" version of equality (i.e. one which is false when either argument does not exist) it is necessary to uncurry equality so that its type becomes ⌜:('a × 'a) → BOOL⌝ and apply it using the system C copula for binary relations.

These two seem to correspond to Grice's intended two kinds of equality insofar as the difference between the two is primarily that one does and the other does not allow that there may be true identities between non-denoting names. However this is achieved in system C by having two quite distinct identity relations only one of which is properly a predicate, and applied through our copula is_s. The other is a propositional function which must be applied without use of the copula (because the copula forces strictness, i.e. forces predication to undefined values to yield falsehoods).

Grice seems to think that the required distinction between equalities can be achieved by the use of his scoping subscripts, but I have not yet been convinced that his is correct in this. Whatever the scope of the predicate, predication to non-denoting names must yield falsehood, so I can't see how any use of subscripts will make "Pegasus = Pegasus" true.

To illustrate the effect of the built in equality the following theorems suffice:

```
Pegasus_eq_lemma1 =
        ⊢ Pegasus = Pegasus
```

Note the lack of copula. This is of course an instance of the theorem:

```
eq_refl_thm =
        ⊢ ∀ x• x = x
```

To get a nice notation for "strong" equality we need to define it as an infix symbol, which requires that the copula be built into the definition. We could use the ≡ symbol, but this usually represents an equivalence relation which is weaker than identity, so we will subscript the = sign with g, suggesting that it is just the same thing applied through our copula.

SML

```
declare_infix (200, "=_s");
```

HOL Constant

```
$=_s : SC → SC → BOOL
────────────────────────────────────────
∀x y:SC• (x =_s y) = (x,y) is_s2 (Uncurry $=)
```

We can now prove:

$$\begin{array}{|l} \mathit{Pegasus\_eq\_lemma2}\ = \\ \qquad \vdash \neg\ \mathit{Pegasus} =_s \mathit{Pegasus} \end{array}$$

And must rest content with the qualified rule:

$$\begin{array}{|l} \boldsymbol{eq_s\_refl\_thm}\ = \\ \qquad \vdash \forall\ x\bullet\ x\ is_s\ \mathit{Existent} \Leftrightarrow x =_s x \end{array}$$

The "strictness" of this equality is expressed by the theorem (which would have made a better definition):

$$\begin{array}{|l} \mathit{eq_s\_strict\_thm}\ = \\ \qquad \vdash \forall\ x\ y\bullet\ x =_s y \Leftrightarrow x\ is_s\ \mathit{Existent} \land y\ is_s\ \mathit{Existent} \land x = y \end{array}$$

The primitive equality in HOL no longer corresponds to Grice's second order definition and the proof of *snd_order_eq_thm* fails.

$$\begin{array}{|l} \boldsymbol{neg\_snd\_order\_eq\_thm}\ = \\ \qquad \vdash \neg\ (\forall\ x\ y\bullet\ x = y \Leftrightarrow (\forall\ P\bullet\ x\ is_s\ P \Leftrightarrow y\ is_s\ P)) \end{array}$$

There are two reasons for this. The first is that there is now more than one possible non-denoting correlate, and Leibniz's formula, applied through the System S copula does not distinguish between them. The second is the presence of values for failing descriptions.

The values for failing descriptions should not be in the range of quantification, so we do need to have special quantifiers in System S, it will then also be necessary to qualify the quantification to get a version of the second order definition which takes account of possibly distinct non-designating names.

This equivalence fails because we now allow (in fact, require) more than one non-designating value in the domain of an interpretation.

However, if the copula is dropped we still get a working definition of a 'weak' equality.

$$\begin{array}{|l} \boldsymbol{snd\_order\_eq\_thm2}\ = \\ \qquad \vdash \forall\ x\ y\bullet\ x = y \Leftrightarrow (\forall\ P\bullet\ P\ x \Leftrightarrow P\ y) \end{array}$$

8 Conclusions

9 Postscript

These are the kinds of thing which *might* happen in future issues of this document:

- Much more detailed discussion and jusification of the method of formalisation, probably split between here and an earlier chapter of the first volume.
- A full discussion of the philosophical issues.
- It would probably be better to rework with a different semantics, with type 'a+'b+'c, separate disjuncts for designating, non-designating and non-correlating expressions, to allow for distinct non-existents and descriptions.

A Theory Listings

A.1 The Theory grice

Parents

rbjmisc

Children

griceS *griceC*

Constants

$\$\models$ $BOOL\ LIST \to BOOL \to BOOL$
$\$$***committal*** $('a \to BOOL) \to ('a \to BOOL) \to BOOL$

Fixity

Right Infix 4: $\models$
Right Infix 100:
committal

Definitions

$\models$ $\vdash \forall\ al\ c\bullet\ (al \models c) \Leftrightarrow \forall_L\ al \Rightarrow c$
committal $\vdash \forall\ P\ \phi\bullet\ P\ committal\ \phi \Leftrightarrow (\forall\ \alpha\bullet\ \phi\ \alpha \Rightarrow P\ \alpha)$

Theorems

ASS $\vdash \forall\ \phi\bullet\ [\phi] \models \phi$
RAA $\vdash \forall\ \phi\ \psi\ \zeta\ \Gamma\bullet\ ([\phi]\ @\ \Gamma \models \psi \wedge \neg\ \psi) \Rightarrow (\Gamma \models \neg\ \phi)$
DN $\vdash \forall\ \phi\bullet\ [\neg\ \neg\ \phi] \models \phi$
$\wedge$***I*** $\vdash \forall\ \phi\ \psi\bullet\ [\phi;\ \psi] \models \phi \wedge \psi$
$\wedge$***E*** $\vdash \forall\ \phi\ \psi\bullet\ ([\phi \wedge \psi] \models \phi) \wedge ([\phi \wedge \psi] \models \psi)$
$\vee$***I*** $\vdash \forall\ \phi\ \psi\bullet\ ([\phi] \models \phi \vee \psi) \wedge ([\phi] \models \psi \vee \phi)$
$\vee$***E*** $\vdash \forall\ \phi\ \psi\ \zeta\ \Gamma\ \Delta\ \Theta$
$\bullet\ ([\phi]\ @\ \Gamma \models \zeta) \wedge ([\psi]\ @\ \Delta \models \zeta) \wedge (\Theta \models \phi \vee \psi)$
$\Rightarrow (\Gamma\ @\ \Delta\ @\ \Theta \models \zeta)$
CP $\vdash \forall\ \phi\ \psi\ \Gamma\bullet\ ([\phi]\ @\ \Gamma \models \psi) \Rightarrow (\Gamma \models \phi \Rightarrow \psi)$
MPP $\vdash \forall\ \phi\ \psi\bullet\ [\phi \Rightarrow \psi;\ \phi] \models \psi$
$\forall$***I*** $\vdash \forall\ \Gamma\ P\bullet\ (\forall\ x\bullet\ \Gamma \models P\ x) \Rightarrow (\Gamma \models (\forall\ x\bullet\ P\ x))$
$\forall$***E*** $\vdash \forall\ \Gamma\ P\ c\bullet\ (\Gamma \models (\forall\ x\bullet\ P\ x)) \Rightarrow (\Gamma \models P\ c)$
$\exists$***I*** $\vdash \forall\ \Gamma\ P\ x\bullet\ (\Gamma \models P\ x) \Rightarrow (\Gamma \models (\exists\ x\bullet\ P\ x))$
$\exists$***E*** $\vdash \forall\ P\ Q\bullet\ (\forall\ x\bullet\ [P\ x] \models Q) \Rightarrow ([\exists\ x\bullet\ P\ x] \models Q)$
P_2 $\vdash \forall\ P\ \phi\bullet\ P\ committal\ \phi \Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \neg\ \neg\ \phi\ \alpha)$
P_3 $\vdash \forall\ P\ \phi\ \psi$
$\bullet\ P\ committal\ \phi \vee P\ committal\ \psi$
$\Rightarrow P\ committal\ (\lambda\ \alpha\bullet\ \phi\ \alpha \wedge \psi\ \alpha)$
P_4 $\vdash \forall\ P\ \phi\ \psi$

$\bullet\ P\ \mathit{committal}\ \phi \land P\ \mathit{committal}\ \psi$

$\quad \Rightarrow P\ \mathit{committal}\ (\lambda\ \alpha \bullet\ \phi\ \alpha \lor \psi\ \alpha)$

***P*_5** $\vdash \forall\ P\ \phi\ \psi$

$\bullet\ P\ \mathit{committal}\ (\lambda\ \alpha \bullet\ \neg\ \phi\ \alpha) \land P\ \mathit{committal}\ \psi$

$\quad \Rightarrow P\ \mathit{committal}\ (\lambda\ \alpha \bullet\ \phi\ \alpha \Rightarrow \psi\ \alpha)$

***P*_6**∀ $\vdash \forall\ P\ \phi$

$\bullet\ (\exists\ \beta \bullet\ P\ \mathit{committal}\ \phi\ \beta)$

$\quad \Rightarrow P\ \mathit{committal}\ (\lambda\ \alpha \bullet\ \forall\ \beta \bullet\ \phi\ \beta\ \alpha)$

P*_6**∃b*** $\vdash \forall\ P\ \phi$

$\bullet\ (\forall\ \beta \bullet\ P\ \mathit{committal}\ \phi\ \beta)$

$\quad \Rightarrow P\ \mathit{committal}\ (\lambda\ \alpha \bullet\ \exists\ \beta \bullet\ \phi\ \beta\ \alpha)$

A.2 The Theory griceC

Parents

grice

Constants

$\bot$	$'a$
$\$\boldsymbol{is_c}$	$'a \to ('a \to BOOL) \to BOOL$
$\$\boldsymbol{is_{c2}}$	$'a \times 'b \to ('a \times 'b \to BOOL) \to BOOL$
$\boldsymbol{Pegasus}$	$'a$
$\boldsymbol{Flier}$	$'a \to BOOL$
$\boldsymbol{NonFlier}$	$'a \to BOOL$
$\boldsymbol{Existent}$	$'a \to BOOL$
$\$\boldsymbol{=_c}$	$'a \to 'a \to BOOL$
$\$\boldsymbol{\iota}$	$('a \to BOOL) \to 'a$

Fixity

Binder: $\boldsymbol{\iota}$

Right Infix 200:

$\boldsymbol{=_c}$

Right Infix 400:

$\boldsymbol{is_c} \quad \boldsymbol{is_{c2}}$

Definitions

$\bot$	$\vdash T$
$\boldsymbol{is_c}$	$\vdash \forall\ p\ t\bullet\ t\ is_c\ p \Leftrightarrow (if\ t = \bot\ then\ F\ else\ p\ t)$
$\boldsymbol{is_{c2}}$	$\vdash \forall\ p\ t$ $\bullet\ t\ is_{c2}\ p$ $\Leftrightarrow (if\ Fst\ t = \bot \lor Snd\ t = \bot\ then\ F\ else\ p\ t)$
$\boldsymbol{Pegasus}$	$\vdash Pegasus = \bot$
$\boldsymbol{Flier}$	$\vdash T$
$\boldsymbol{NonFlier}$	$\vdash \forall\ x\bullet\ NonFlier\ x \Leftrightarrow \neg\ Flier\ x$
$\boldsymbol{Existent}$	$\vdash \forall\ x\bullet\ Existent\ x \Leftrightarrow T$
$\boldsymbol{=_c}$	$\vdash \forall\ x\ y\bullet\ x =_c y \Leftrightarrow (x,\ y)\ is_{c2}\ Uncurry\ \$=$
$\boldsymbol{\iota}$	$\vdash \forall\ \phi$ $\bullet\ \$\iota\ \phi$ $= (\epsilon\ \alpha$ $\bullet\ \alpha\ is_c\ \phi \land (\forall\ \beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta = \alpha)$ $\lor \neg\ (\exists\ \gamma\bullet\ \gamma\ is_c\ \phi \land (\forall\ \beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta = \gamma))$ $\land\ \alpha = \bot)$

Theorems

Pegasus_lemma_1

$\vdash Pegasus\ is_c\ Flier \Leftrightarrow F$

Pegasus_lemma_2

$\vdash Pegasus\ is_c\ (\lambda\ x\bullet\ \neg\ Flier\ x) \Leftrightarrow F$

Pegasus_lemma_3

$\vdash \neg\ Pegasus\ is_c\ Flier$

Pegasus_lemma_4

$\vdash \neg\ Pegasus\ is_c\ (\lambda\ x\bullet\ \neg\ Flier\ x)$

Pegasus_lemma_5

$\vdash Pegasus\ is_c\ NonFlier \Leftrightarrow F$

Existent_lemma_1

$\vdash \forall\ P\ x\bullet\ \neg\ x\ is_c\ Existent \Rightarrow \neg\ x\ is_c\ P$

∃*_lemma_1* $\vdash \forall\ P\bullet\ Pegasus\ is_c\ P \Rightarrow (\exists\ x\bullet\ x\ is_c\ P)$

∀*_lemma_1* $\vdash \forall\ P\bullet\ (\forall\ x\bullet\ x\ is_c\ P) \Rightarrow Pegasus\ is_c\ P$

E1 $\vdash \forall\ \phi\ \alpha\bullet\ \alpha\ is_c\ \phi \Rightarrow \alpha\ is_c\ Existent$

Pegasus_eq_lemma1

$\vdash Pegasus\ =\ Pegasus$

eq_refl_thm $\vdash \forall\ x\bullet\ x\ =\ x$

Pegasus_eq_lemma2

$\vdash \neg\ Pegasus\ =_c\ Pegasus$

eq$_c$_refl_thm

$\vdash \forall\ x\bullet\ x\ is_c\ Existent \Leftrightarrow x\ =_c\ x$

eq$_c$_strict_thm

$\vdash \forall\ x\ y$

$\bullet\ x\ =_c\ y \Leftrightarrow x\ is_c\ Existent \wedge y\ is_c\ Existent \wedge x\ =\ y$

snd_order_eq_thm

$\vdash \forall\ x\ y\bullet\ x\ =\ y \Leftrightarrow (\forall\ P\bullet\ x\ is_c\ P \Leftrightarrow y\ is_c\ P)$

snd_order_eq_thm2

$\vdash \forall\ x\ y\bullet\ x\ =\ y \Leftrightarrow (\forall\ P\bullet\ P\ x \Leftrightarrow P\ y)$

ι_thm1 $\vdash \forall\ \gamma\bullet\ \gamma\ is_c\ \phi \wedge (\forall\ \beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta\ =\ \gamma) \Rightarrow \$\iota\ \phi\ =\ \gamma$

ι_thm2 $\vdash \neg\ (\exists\ \gamma\bullet\ \gamma\ is_c\ \phi \wedge (\forall\ \beta\bullet\ \beta\ is_c\ \phi \Rightarrow \beta\ =\ \gamma))$

$\Rightarrow \$\iota\ \phi\ =\ \bot$

A.3 The Theory griceS

Parents

grice

Constants

$\boldsymbol{denotatum}$	$'a \to BOOL$
$\bot\_\boldsymbol{corr}$	$'a$
$\bot$	SC
$\boldsymbol{IsCorr}$	$SC \to BOOL$
$\boldsymbol{Corr}$	$SC \to {'a}$
$\boldsymbol{IsDen}$	$SC \to BOOL$
$\$\boldsymbol{is_s}$	$SC \to ('a \to BOOL) \to BOOL$
$\$\boldsymbol{is_{s2}}$	$SC \times SC \to ('a \times 'a \to BOOL) \to BOOL$
$\boldsymbol{Pegasus}$	SC
$\boldsymbol{Flier}$	$'a \to BOOL$
$\boldsymbol{NonFlier}$	$'a \to BOOL$
$\boldsymbol{Existent}$	$'a \to BOOL$
$\$\forall_{\boldsymbol{s}}$	$(SC \to BOOL) \to BOOL$
$\$\exists_{\boldsymbol{s}}$	$(SC \to BOOL) \to BOOL$
$\boldsymbol{Existent_s}$	$SC \to BOOL$
$\$=_{\boldsymbol{s}}$	$SC \to SC \to BOOL$

Type Abbreviations

$\boldsymbol{SC}$	SC

Fixity

Binder: $\forall_{\boldsymbol{s}}$ $\exists_{\boldsymbol{s}}$
Right Infix 200:
$=_{\boldsymbol{s}}$
Right Infix 400:
$\boldsymbol{is_s}$ $\boldsymbol{is_{s2}}$

Definitions

$\boldsymbol{denotatum}$	$\vdash \exists\ x \bullet\ \neg\ denotatum\ x$
$\bot\_\boldsymbol{corr}$	$\vdash \neg\ denotatum\ \bot\_corr$
$\bot$	$\vdash \bot = InL\ \bot\_corr$
$\boldsymbol{IsCorr}$	$\vdash \forall\ v \bullet\ IsCorr\ v \Leftrightarrow IsL\ v$
$\boldsymbol{Corr}$	$\vdash \forall\ v \bullet\ Corr\ v = OutL\ v$
$\boldsymbol{IsDen}$	$\vdash \forall\ v \bullet\ IsDen\ v \Leftrightarrow IsCorr\ v \wedge denotatum\ (Corr\ v)$
$\boldsymbol{is_s}$	$\vdash \forall\ p\ t \bullet\ t\ is_s\ p \Leftrightarrow (if\ IsDen\ t\ then\ p\ (Corr\ t)\ else\ F)$
$\boldsymbol{is_{s2}}$	$\vdash \forall\ p\ t$ $\bullet\ t\ is_{s2}\ p$ $\Leftrightarrow (if\ IsDen\ (Fst\ t) \wedge IsDen\ (Snd\ t)$ $then\ p\ (Corr\ (Fst\ t),\ Corr\ (Snd\ t))$

$\qquad\qquad\mathit{else}\ F)$

Pegasus $\vdash \mathit{IsCorr}\ \mathit{Pegasus} \land \lnot\ \mathit{IsDen}\ \mathit{Pegasus}$

Flier $\vdash T$

NonFlier $\vdash \forall\ x\bullet\ \mathit{NonFlier}\ x \Leftrightarrow \lnot\ \mathit{Flier}\ x$

Existent $\vdash \forall\ x\bullet\ \mathit{Existent}\ x \Leftrightarrow T$

$\forall_{\boldsymbol{s}}$ $\vdash \forall\ f\bullet\ \$\forall_s\ f \Leftrightarrow (\forall\ x\bullet\ \mathit{IsCorr}\ x \Rightarrow f\ x)$

$\exists_{\boldsymbol{s}}$ $\vdash \forall\ f\bullet\ \$\exists_s\ f \Leftrightarrow (\exists\ x\bullet\ \mathit{IsCorr}\ x \land f\ x)$

Existent$_{\boldsymbol{s}}$ $\vdash \forall\ x\bullet\ \mathit{Existent}_s\ x \Leftrightarrow (\exists_s\ y\bullet\ x = y)$

$=_{\boldsymbol{s}}$ $\vdash \forall\ x\ y\bullet\ x =_s y \Leftrightarrow (x,\ y)\ \mathit{is}_{s2}\ \mathit{Uncurry}\ \$=$

Theorems

Pegasus_lemma_1

$\vdash \mathit{Pegasus}\ \mathit{is}_s\ \mathit{Flier} \Leftrightarrow F$

Pegasus_lemma_2

$\vdash \mathit{Pegasus}\ \mathit{is}_s\ (\lambda\ x\bullet\ \lnot\ \mathit{Flier}\ x) \Leftrightarrow F$

Pegasus_lemma_3

$\vdash \lnot\ \mathit{Pegasus}\ \mathit{is}_s\ \mathit{Flier}$

Pegasus_lemma_4

$\vdash \lnot\ \mathit{Pegasus}\ \mathit{is}_s\ (\lambda\ x\bullet\ \lnot\ \mathit{Flier}\ x)$

Pegasus_lemma_5

$\vdash \mathit{Pegasus}\ \mathit{is}_s\ \mathit{NonFlier} \Leftrightarrow F$

Existent_lemma_1

$\vdash \forall\ P\ x\bullet\ \lnot\ x\ \mathit{is}_s\ \mathit{Existent} \Rightarrow \lnot\ x\ \mathit{is}_s\ P$

$\exists$***_lemma_1*** $\vdash \forall\ P\bullet\ \mathit{Pegasus}\ \mathit{is}_s\ P \Rightarrow (\exists\ x\bullet\ x\ \mathit{is}_s\ P)$

$\forall$***_lemma_1*** $\vdash \forall\ P\bullet\ (\forall\ x\bullet\ x\ \mathit{is}_s\ P) \Rightarrow \mathit{Pegasus}\ \mathit{is}_s\ P$

$\exists$***_not_Corr_thm***

$\vdash \exists\ x\bullet\ \lnot\ \mathit{IsCorr}\ x$

$\forall\boldsymbol{I_s}$ $\vdash \forall\ \Gamma\ P$

$\quad\bullet\ (\forall\ x\bullet\ \Gamma \models \mathit{Existent}_s\ x \land P\ x) \Rightarrow (\Gamma \models (\forall_s\ x\bullet\ P\ x))$

$\forall\boldsymbol{E_s}$ $\vdash \forall\ \Gamma\ P\ c$

$\quad\bullet\ (\Gamma \models \mathit{Existent}_s\ c \land (\forall_s\ x\bullet\ P\ x)) \Rightarrow (\Gamma \models P\ c)$

$\exists\boldsymbol{I_s}$ $\vdash \forall\ \Gamma\ P\ x$

$\quad\bullet\ (\Gamma \models \mathit{Existent}_s\ x \land P\ x) \Rightarrow (\Gamma \models (\exists_s\ x\bullet\ P\ x))$

$\exists\boldsymbol{E_s}$ $\vdash \forall\ P\ Q\ c$

$\quad\bullet\ (\forall\ x\bullet\ [\mathit{Existent}_s\ x \land P\ x] \models Q)$

$\qquad\Rightarrow ([\exists_s\ x\bullet\ P\ x] \models Q)$

***E*1** $\vdash \forall\ \phi\ \alpha\bullet\ \alpha\ \mathit{is}_s\ \phi \Rightarrow \mathit{Existent}\ \alpha$

***Pegasus_eq_lemma*1**

$\vdash \mathit{Pegasus} = \mathit{Pegasus}$

eq_refl_thm $\vdash \forall\ x\bullet\ x = x$

***Pegasus_eq_lemma*2**

$\vdash \lnot\ \mathit{Pegasus} =_s \mathit{Pegasus}$

eq$_{\boldsymbol{s}}$***_refl_thm***

$\vdash \forall\ x\bullet\ x\ \mathit{is}_s\ \mathit{Existent} \Leftrightarrow x =_s x$

eq$_{\boldsymbol{s}}$***_strict_thm***

$\vdash \forall\ x\ y$

$\quad\bullet\ x =_s y \Leftrightarrow x\ \mathit{is}_s\ \mathit{Existent} \land y\ \mathit{is}_s\ \mathit{Existent} \land x = y$

neg_snd_order_eq_thm

$\vdash \lnot\ (\forall\ x\ y\bullet\ x = y \Leftrightarrow (\forall\ P\bullet\ x\ \mathit{is}_s\ P \Leftrightarrow y\ \mathit{is}_s\ P))$

***snd_order_eq_thm*2**

$\vdash \forall\ x\ y\bullet\ x = y \Leftrightarrow (\forall\ P\bullet\ P\ x \Leftrightarrow P\ y)$

A.4 *

Bibliography

[1] Alonzo Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–, 1940.

[2] Davidson and Hintikka, editors. *Words and Objections: Essays on the work of W.V. Quine.* Reidel Publishing. 1969.

[3] Richard E. Grandy and Richard Warner, editors. *Philosophical Grounds of Rationality: Intentions, Categories, Ends.* Clarendon Press. 1986.

[4] H.P. Grice. Vacuous Names. In Davidson and Hintikka [2].

[5] H.P. Grice. *Studies in the Way of Words.* Harvard University Press. 1989.

[6] Roger Bishop Jones. Analyses of Analysis: Part I - Introduction. *RBJones.com*, 2009-. http://www.rbjones.com/rbjpub/pp/doc/t029.pdf.

[7] Roger Bishop Jones. *Analyses of Analysis: Part II - Synthetic Analysis.* RBJones.com. 2009. http://www.rbjones.com/rbjpub/pp/doc/b003.pdf.

[8] Benson Mates. *Elementary Logic.* Oxford University Press. 1965.

[9] George Myro. Identity and Time. In Grandy and Warner [3].

[10] Bertrand Russell. On Denoting. [11].

[11] Bertrand Russell. *Logic and Knowledge.* Routledge. 1956.

A.5 *

Index