

Quine on necessity, extensionality, and essentialism

Jeff Speaks

March 18, 2005

1	Three kinds of ‘necessity’ operator	1
2	Extensionality and referential opacity	2
3	The connection between failure of extensionality and referential opacity	2
4	The validation of ‘Necessarily’	3
5	Quantifying in to referentially opaque contexts	5
6	‘Nec’ as a sentence operator and Aristotelian essentialism	5

So far we have discussed Quine’s arguments, first, against the explanation of necessity and the a priori in terms of convention, and, second, against the intelligibility of the analytic/synthetic distinction. If one takes necessity to be explained in terms of analyticity, the latter already counts against the intelligibility of the necessary/contingent distinction. But Quine (in, among other places, “Three Grades of Modal Involvement”) also presented some independent arguments (independent, that is, of considerations about analyticity’s relation to necessity) against the unrestricted use of a ‘necessity’ operator.

1 Three kinds of ‘necessity’ operator

Quine begins by distinguishing three different kinds of ‘necessity’ operator we might admit into our language. These three kinds of operators correspond to three different sorts of necessary/contingent distinction. The idea is that we can test the intelligibility of these three levels of the necessary/contingent distinction by testing the intelligibility of the corresponding ‘necessity’ operators.

The three kinds of ‘necessity’ operator are:

1. ‘Necessity’ as a semantical predicate (i.e., predicate of sentences), as in “Nec ‘9>5’.”
2. ‘Necessity’ as a statement operator (i.e., not a predicate of sentences, but an expression which combines with a complete sentence to form another complete sentence), as in “Nec 9>5.”
3. ‘Necessity’ as a sentence operator (similar to the use of ‘Nec’ as a statement operator, with the addition that the operator can also be combined with open sentences containing free variables which can then be bound by quantifiers), as in “ $\exists x$ (Nec $x > 5$)”

The question is then whether we can make sense of any, or all, of these three necessity operators.

2 Extensionality and referential opacity

To understand Quine's discussion, we'll need to understand a few of his technical terms.

The first is the distinction between *purely referential* and *non-referential* occurrences of singular terms in sentences. Quine gives a rough definition of the former: an occurrence is purely referential iff the singular term, in that occurrence, serves simply to refer to its object. A good test of whether an occurrence of a singular term in a sentence is purely referential or not, Quine suggests, is whether we can always substitute other singular terms with the same reference without change of truth-value.

Related to this is the distinction between *referentially opaque* and *referentially transparent* contexts in sentences. This is a distinction between two different kinds of contexts in which sentences can occur within larger sentences. A context in which a sentence can occur as part of a larger sentence is referentially opaque iff it can change a purely referential occurrence of a singular term into a non-referential one. The "referentially opaque context par excellence," says Quine, is quotation, since while the occurrence of 'Napoleon' in

Napoleon escaped from Elba.

is purely referential, the occurrence of 'Napoleon' in

"Napoleon escaped from Elba" is my favorite sentence.

is not.

Finally, we have the distinction between *extensional* and *non-extensional* contexts. Again, this is a distinction between two different kinds of contexts in which sentences can occur within larger sentences. Such a context is extensional iff it is truth-functional (if and only if, that is, sentences with the same truth-value may be substituted in that context without change in the truth-value of the larger sentence). Quotation creates non-extensional contexts, as does 'Necessarily' used as a statement operator, since although

Necessarily, $2+2=4$.

is true, and ' $2+2=4$ ' has the same truth-value as 'Napoleon escaped from Elba',

Necessarily, Napoleon escaped from Elba.

is false.

3 The connection between failure of extensionality and referential opacity

The phenomena of extensionality and referential opacity seem to be connected in some way; it is unsurprising, for example, that quotational contexts are examples of both non-extensionality and referential opacity. On p. 161, Quine presents an argument that the two kinds of contexts are related. The argument shows that certain kinds of non-extensional contexts must also be referentially opaque.

To begin, note that although non-extensional contexts do not permit substitution *salva veritate* of sentences with the same truth-value, there is a distinction between those sentences which do, and those which do not, permit substitution *salva veritate* of sentences which are logically equivalent. Why ‘Necessarily’, used as a statement operator, seems to fall into the former class.

Using these facts, Quine presents the following argument that ‘Necessarily’ creates referentially opaque contexts:

0. ‘Necessarily’ does not create a referentially opaque context. (Assume for reductio)
1. ‘Necessarily’ creates a non-extensional context.
2. ‘Necessarily’ permits substitution *salva veritate* of logically equivalent sentences. (If sentences S and S' are logical equivalents, then \ulcorner Necessarily, $S\urcorner$ is true iff \ulcorner Necessarily, $S'\urcorner$ is.)
3. Let ‘ p ’ be some true sentence such that ‘Necessarily, p ’ is true.
4. ‘ p ’ is logically equivalent to ‘ $\hat{x} (X=\emptyset \ \& \ p) = \emptyset$ ’
5. ‘Necessarily $\hat{x} (X=\emptyset \ \& \ p) = \emptyset$ ’ is true. (follows from 2,3,4)
6. Let ‘ q ’ be some sentence which is true (and hence has the same truth value as ‘ p ’).
7. $\hat{x} (X=\emptyset \ \& \ p) = \hat{x} (X=\emptyset \ \& \ q)$ (follows from 6)
8. ‘Necessarily $\hat{x} (X=\emptyset \ \& \ q) = \emptyset$ ’ is true. (follows from 0,5,7)
9. ‘ q ’ is logically equivalent to ‘ $\hat{x} (X=\emptyset \ \& \ q) = \emptyset$ ’
10. ‘Necesssarily, q ’ is true. (follows from 2,8,9)

So, starting only from the assumptions that ‘Necessarily, p ’ is true and that ‘ p ’ and ‘ q ’ have the same truth-value, we were able to show that ‘Necessarily, q ’ must also be true. But the fact that we can do this for any propositions which have the same truth-value contradicts our claim that ‘Necessarily’ creates a non-extensional context. But it does create a non-extensional context. So one of our other assumptions must be incorrect — in particular, either (0) or (2) must be incorrect. But (2) is clearly true. So (0) must be incorrect, and ‘Necessarily’ creates referentially opaque contexts.

This argument shows that any operators which permit substitution of logical equivalents and are non-extensional create referentially opaque contexts. Since necessity is such an operator, the arguments shows that it creates referentially opaque contexts.

4 The validation of ‘Necessarily’

Why Quine thinks we need something like ‘Necessarily’ in its use as a semantic predicate.

An extension of this reasoning to validate uses of ‘Necessarily’ as a statement operator. The idea that sentences like

Necessarily, $9 > 5$.

can be understood as saying the same thing as

‘ $9 > 5$ ’ is necessary.

Once we have justified the use of ‘Necessarily’ as a statement operator, Quine notes that it is a natural step to use it as a sentence operator as well:

“Having adopted . . . negation as applicable to statements, one applies it without thought to open sentences as well: sentences containing free variables ripe for quantification. Thus we can write not only ‘ \neg (Socrates is mortal)’ but also ‘ $\neg(x \text{ is mortal})$ ’, from which, by quantification and further negation, we have . . . ‘ $(\exists x) (x \text{ is mortal})$ ’. With negation this is as it should be. As long as ‘nec’ is used as a statement operator, on a par with negation, the analogous course suggests itself again . . .” (169-170)

But Quine thinks that this move — from treating ‘Necessarily’ as a statement operator to treating it as a sentence operator — is far from harmless:

“. . . it is a drastic one, for it suddenly obstructs the earlier expedient of translation into terms of ‘Nec’ as a semantical predicate. . . we cannot construe

(45) $\text{nec } (x > 5)$

correspondingly as

(46) $\text{Nec } 'x > 5'$. ”

He explains why later on the page:

“whereas (45) is an open sentence with free ‘ x ’, (46) has no corresponding generality; (46) is simply a statement *about* a specific open sentence. For, it must be remembered that ‘ $x > 5$ ’ in quotation marks is a name of the specific quoted expression, with fixed letter ‘ x .’ The ‘ x ’ in (46) cannot be reached by quantifier. To write:

(47) $\exists x (\text{Nec } 'x > 5')$

is like writing

(48) $\exists x (\text{Socrates is mortal})$;

the quantifier is followed by no germane occurrence of its variable. In a word, necessity as a sentence operator does not go over into terms of necessity as a semantical predicate.” (170)

The conclusion at this stage is that we cannot justify the use of ‘Nec’ as a sentence operator by showing how we can translate such uses into uses of ‘Nec’ as a semantical predicate. This leaves open the question of whether uses of ‘Nec’ as a sentence operator are intelligible.

5 Quantifying in to referentially opaque contexts

We now get an argument that it is not. We know from the argument discussed above that, since ‘Necessarily’ is non-extensional and permits substitution of logical equivalents, it must create referentially opaque contexts. But this, Quine indicates, is enough to cast doubt on uses of ‘Necessarily’ as a sentence operator:

“acceptance of necessity as a sentence operator implies an attitude quite opposite to our earlier one, which was that ‘nec’ as a statement operator is referentially opaque. For, one would clearly have no business quantifying into a referentially opaque context; witness (47) above. We can reasonably infer ‘ $(\exists x) \text{ nec } (x > 5)$ ’ from ‘ $\text{nec } (9 > 5)$ ’ only if we regard the latter as telling us something about the *object* 9, a number, viz. that it necessarily exceeds 5. If ‘ $\text{nec } (\dots > 5)$ ’ can turn out true or false ‘of’ the number 9 depending merely on how that number is referred to (as the falsity of [‘ $\text{nec } (\text{the number of planets} > 5)$ ’] suggests), then evidently ‘ $\text{nec } (x > 5)$ ’ expresses no genuine condition on objects of any kind. If the occurrence of ‘9’ in ‘ $\text{nec}(9 > 5)$ ’ is not purely referential, then putting ‘ x ’ for ‘9’ in ‘ $\text{nec } (9 > 5)$ ’ makes no more sense than putting ‘ x ’ for ‘nine’ within the context ‘canine.’” (170-171)

6 ‘Nec’ as a sentence operator and Aristotelian essentialism