Chapter 7

The theory at work

In this chapter, the singularity theory is put to work on a number of paradoxes that have intrinsic interest of their own. These include a 'transfinite' paradox of denotation which shows the need to accommodate limit ordinals, versions of the Liar that will prove significant for our later discussions of revenge (in Chapters 8 and 9), and new versions of the definability paradoxes and the Russell paradox that do not involve self-reference or circularity.

7.1 A transfinite paradox of denotation

We now consider in more detail the transfinite case of denotation that was introduced in the previous chapter. Although the focus here is on denotation, parallel paradoxes can be constructed for the notions of extension and truth.

We suppose that, for any n, a_n, b_n and c_n are written on the board in room n, and that the token d is written elsewhere.

(d) the sum of the numbers denoted by c_0, c_1, c_2, ...

(a_0) 0
(b_0) the square of 0.
(c_0) the sum of the numbers denoted by expressions in room 0.

(a_1) 0
(b_1) the square of 0.
(c_1) the sum of the numbers denoted by c_0 and the expressions in room 1.
.

.
.

(a_{n+1}) 0
(b_{n+1}) the square of 0.
(c_{n+1}) the sum of the numbers denoted by c_n and the expressions in room n+1.
.
.
.

As we observed in the previous chapter, c_0 is a reflection-free expression (the infinite branch of c_0's pruned_0 tree is a loop). And it's straightforward to check that c_1 is 1-reflective, c_2 is 2-reflective, and in general, c_n is n-reflective. For example, consider the pruned_1 tree for c_1.

```
                           <type(c_1),c_c1,c_c1>
                      /         /        \          \
                    /         /            \          \
<type(c_0),c_c0, c_c1>   <type(a_0)>   <type(b_0)>   <type(c_1),c_c1,c_c1>
                                                          /          |          \        .
                                                        /            |            \          .
                                  <type(c_0),c_c0, c_c1>   <type(a_0)>   <type(b_0)>        .
```

Here the node $<type(c_0),c_{c0},c_{c1}>$ is a reflective 1-representation, and the only infinite branch is a loop. So c_1 is 1-reflective.

Now consider d, and the pruned_n tree for d for any n. None of the infinite branches of d's pruned_n tree are loops or chains: not loops, because the primary representation of d does not repeat on any infinite branch, and no secondary representation of d makes an appearance; and not chains, because every infinite branch has repeating nodes (the primary representation of c_m, for any m>n, repeats on d's pruned_n tree). So d is not n-reflective, for any finite ordinal n.

But it's straightforward to show that d is $<\omega$-reflective, where ω is the first infinite ordinal. It is clear that d's primary tree contains a reflective n-representation, for any n such that

$1 \leq n < \alpha$. The nodes at the second tier of d's primary tree are $<type(c_o),c_{co},c_d>$, $<type(c_1),c_{c1},c_d>$, … , $<type(c_1),c_{c1},c_d>$, … , and each of these nodes is a reflective n-representation, for $1 \leq n < \omega$. And it is clear that for each such reflective n-representation, there is a reflective m-representation on d's primary tree, where $1 \leq n < m < \alpha$. Now consider the $pruned_{<\alpha}$ tree for d:

```
                                <type(d),cd,cd>
                    /                 |          …    \                     …
                   /                  |            …    \                       …
<type(c0),cc0, ccd>        <type(c1),cc1, ccd>  …   <type(cn),ccn, ccd>       …
```

Since there are no infinite branches, clause (iii) of the definition of an $<\alpha$-reflective expression is vacuously satisfied. So d is $<\omega$-reflective, and an ω-expression.

There are variants of d which are also ω-reflective - for example:

$(d^{/})$ the sum of the numbers denoted by c_1, c_3, c_5, ...c_{2n+1} ... .

Notice that $d^{/}$'s $pruned_{<\omega}$ tree does not contain a reflective n-representation for every n, only for some. But it is still true that for each reflective n-representation on d's primary tree, there is a reflective m-representation on the tree, where $1 \leq n < m < \omega$.

Given our conveniently chosen expressions a_0, a_1, ... , b_0, b_1, ... , we can easily establish values for c_0, c_1, c_2, ... , and for d. The determination tree for c_0 is its $pruned_0$ tree:

```
                 <type(c0),cc0, cc0>
                /          |             \
               /           |              \
<type(a0)>      <type(b0)>     <type(c0),cc0, cc0>
                                   /       |         .
                                  /        |           .
                     <type(a0)>      <type(b0)>         .
```

This determination tree indicates that c_0 is a singularity of the occurrence of '$denotes_{c0}$' in c_0. So the value for c_0 is determined from the members of c_0's determination set other than c_0. The 0-

expressions a_0 and b_0 each denote$_{c0}$ 0. So the procedure framed by the determination-tree for c_0 yields the value 0 for c_0. The determination trees for the expressions c_1, c_2, ... c_n, … frame procedures that determine the value 0 for each of them.

The determination tree for d is its pruned$_{<\omega}$ tree. The terminal nodes (at the second tier) are reflective representations of the form <type(c_n),c_{cn}, c_{cd}>, and the value of each of these nodes is in turn given by the determination tree for c_n. The value associated with each terminal node is 0, and so the determination tree for d yields the value 0.

7.2 The Truth-Teller, a Curry sentence, loops

7.2.1 The Truth-Teller and the Curry Paradox

Suppose I write on the board in room 213 just one sentence:

(T) The sentence on the board in room 213 is true.

Then I have produced one version of the Truth-Teller – a sentence that says of itself that it is true. The primary representation of T is <type(T),c_T,c_T>, and T's primary tree consists of a single infinite branch. T is identified as a singularity of 'true' in T. Once T is removed from the extension of 'true' in T, T may be reflectively evaluated as false – that is, false$_{rT}$. This is the natural result: since T is pathological, T isn't true -- but this is the negation of T, and since the negation of T is true, T is false.[1]

Curry's paradox is related to the Truth-Teller. The Curry reasoning seems to allow us to prove any claim we like – say, the claim that 2+2=5. Consider the following sentence:

(U) If U is true then 2+2=5.

We argue as follows: Assume

(a) U is true.

From (a) and the substitutivity of identicals, we obtain

(b) 'If U is true, then 2+2=5' is true.

By the truth-schema,

(c) 'If U is true, then 2+2=5' is true iff if U is true then 2+2=5.

From (b) and (c), it follows by truth functional logic that

(d) If U is true then 2+2=5.

From (a) and (d), by modus ponens, we infer

(e) 2+2=5.

Thus far, we have inferred (e) on the basis of (a) alone. So we obtain, by conditional introduction,

(f) If U is true then 2+2=5.

From (c) and (f), by truth functional logic, it follows that

(g) 'If U is true then 2+2=5' is true.

From (g) and the substitutivity of identicals, we infer

(h) U is true.

And now from (d) and (h), by modus ponens, we reach the conclusion

(i) 2+2=5.

This version of the Curry paradox is easily converted to an 'empirical' version.[2] Suppose I see the sentence '2+2=5' written on the board next door, and proceed to write what I take to be a trivial truth:

(U) If the sentence on the board in Caldwell 213 is true then 2+2=5.

But I'm in Caldwell 213 – and so U is a Curry sentence.

Let c_U be the context of utterance of the Curry sentence. Then the occurrence of 'true' in U is represented by '$true_{cU}$', U's associated schema is the c_U-schema, and the primary representation of U is <type(U),c_U,c_U>. Essential use of the c_U-schema is made in the step from (b) to (c). This is the step where the argument breaks down. When we instantiate the c_U-schema to U, we obtain a false instance, as we will now see.

The antecedent of U – call it V – contains an occurrence of 'true' that is, as we have seen, represented by '$truec_U$'. And V shares with U its associated schema, since V is a component of U. So the primary representation of V is <type(V),c_U,c_U>. The determination set of U is composed of its truth functional components (see 9.1.2). So the $pruned_0$ tree for U is as follows:

```
            <type(U),c_U,c_U>
               /          \
              /            \
   <type(V),c_U,c_U>      <type('2+2=5')
              |
              |
   <type(U),c_U,c_U>
     .            \
    .              \
   .              <type('2+2=5')
```

The primary representation of U repeats on the infinite branch, and so U is a key singularity of 'true' in U. So we can reflectively evaluate U by excluding it from the scope of 'true' in U. The result of the exclusion is that the antecedent V is false, and so the conditional U is true, when reflectively evaluated.[3]

When we instantiate the c_U-schema to U, we obtain this instance:

U is $true_{cU}$ iff if U is $true_{cU}$ then 2+2=5.

The left-hand-side is false, because U is a singularity of '$true_{cU}$'. The right-hand-side is true, because it is a conditional with a false antecedent. So this instance of the schema is false. The Curry argument makes illegitimate use of the truth-schema, and step (c) is false. Again, paradox is resolved by the identification and exclusion of singularities.

It is perhaps worth noting that the Curry paradox poses special problems for paracomplete theories of truth (e.g. Field's) and for paraconsistent theories (e.g. Priest's).[4] In contrast, the singularity theory does not depart from classical logic, and treats Curry's paradox in just the same way in which it treats all the paradoxes, by the identification and exclusion of singularities.

7.2.2 Loops

Consider the following pair of sentences, written on the board in room 213:

(A) B is not true.

(B) A is not true.

This pair can seem to give rise to inescapable pathology. Either we assign the same value to A and B, or we don't. If we assign the same value, the result is inconsistency. If we assign different values, the result is indeterminacy, since there is no reason to prefer one such assignment over another.

This is a simple loop, and a version of what is sometimes called the Open Pair paradox.[5] The singularity treatment runs as follows. A's primary tree looks like this:

<type(A),c_A,c_A>
|
<type(B),c_B,c_A>
|
<type(A),c_A,c_B>
|
<type(B),c_B,c_A>
|
<type(A),c_A,c_B>
.
.
.

Secondary representations of A and B appear on this infinite branch. In accordance with Symmetry, then, A and B are both pathological, and each is a singularity of the occurrences of 'true' in A and B. A's primary tree is also its determination tree, and a reflective value for A – namely, *true* -- is established by excluding the singularity B from 'true' in A (see 6.5). The same value for B is established by B's determination tree.

This treatment of the Open Pair maintains symmetry, consistency, and determinacy. A is pathological – it cannot be evaluated by the c_A-schema (or the c_B-schema), and so it is not $true_{cA}$ (and not $true_{cB}$). B is likewise pathological – it cannot be evaluated by the c_B-schema (or the c_A-schema), and so it is not $true_{cB}$ (and not $true_{cA}$). And A and B are both true when reflectively evaluated. We might have the intuition that if one is true, the other is not true – that they must have opposite values. But this intuition is also accommodated: A is true on reflection just because B is not $true_{cB}$, and B is true on reflection just because A is not $true_{cA}$.

For a more sophisticated 'empirical' Liar loop, consider the following case adapted from Kripke.[6] Suppose Nixon says

(1) All of Dean's utterances about Watergate are not true.

And suppose Dean says

(2) Everything Nixon says about Watergate is not true.

Each of these utterances includes the other in its scope. This need not mean that we are unable to assign truth values to (1) and (2). Suppose Dean has made at least one true statement about Watergate, expressed in the language *L*, the fragment of English free of semantic terms. Then, according to the singularity theory, (1) is a *0-expression*, since its value (false) is determined by a sentence of *L* (see 9.3). If, further, every other statement of Nixon's about Watergate is a false sentence of *L*, then (2) is also a (true) 0-expression.[7]

But now suppose that all Watergate-related statements made by Nixon and Dean, other than (1) and (2), are false sentences of *L*. Recall the notion of a pruned$_0$ tree from 6.5. Each of the pruned$_0$ trees for (1) and (2) contain a single infinite branch. For example, the infinite branch of (1)'s pruned$_0$ tree looks like this:

$$\langle \text{type}(1), c_1, c_1 \rangle$$
$$|$$
$$\langle \text{type}(2), c_2, c_1 \rangle$$
$$|$$
$$\langle \text{type}(1), c_1, c_2 \rangle$$
$$|$$
$$\langle \text{type}(2), c_2, c_1 \rangle$$
$$|$$
$$\langle \text{type}(1), c_1, c_2 \rangle$$
.
.
.

Secondary representation of (1) and (2) repeat on this infinite branch, so (1) and (2) are looped – they are pathological reflection-free expressions. (2) is identified as a key singularity of 'true' in (1), and (1) is identified as a key singularity of 'true' in (2) (again, see 6.5). The determination tree for (1) – its pruned$_0$ tree – displays what we need to evaluate (1). Each finite branch terminates in an untrue sentence of *L*. Via the infinite branch, we can identify (2) as a key singularity of 'true' in (1), and exclude it from the scope of 'true' in (1). So all of the members of (1)'s determination set are excluded from the scope of 'true' in (1). So all of Dean's utterances are indeed not true_{c1}, and that's what (1) says – so we can reflectively evaluate (1) as true. This matches our intuitions – all of Dean's statements about Watergate are not true, because they're either false or pathological. The situation with (2) is exactly symmetrical.[8]

7.3 New paradoxes without circularity

The paradoxes of definability - Berry's paradox, Richard's paradox, and König's paradox - all exhibit some form of circularity, or self-reference understood in a suitably broad sense. Consider Berry's paradox, for example, generated by the expressions "the least positive integer which is not denoted by an expression of English containing fewer than thirty five syllables". Observe that the Berry phrase involves quantification over a certain domain of English phrases which contains the Berry phrase itself. Here is the circularity - or self-reference, in the sense that the Berry phrase makes indirect reference to itself. In the case of C, the self-reference and circularity is even more explicit.

Similarly, all the standard set-theoretical paradoxes exhibit circularity and self-reference, broadly construed. Standard versions of Russell's paradox turn on self-membership and circularity. For example, we consider the predicate 'non-self-membered extension', and ask whether or not the extension of this very predicate is self-membered. (The circularity exhibited by the predicate P that generates the simple Russell is immediately obvious.) The set-theoeretical paradoxes due to Burali-Forti and Cantor exhibit circularity too. Russell thought that circularity was the culprit in all of the paradoxes, semantic and set-theoretical, and proposed the Vicious Circle Principle to deal with it.

7.3.1 A definability paradox without circularity

We can certainly say that there are pathological denoting expressions whose pathology does not turn on circularity or self-reference. Consider the following infinite chain of expressions:

A_1. The positive integer denoted by A_2.

A_2. The positive integer denoted by A_3.

.
.
.

A_n The positive integer denoted by A_{n+1}.

.
.
.

But we cannot be said to have a paradox or antinomy here -- no contradiction is forced upon us.

Let us work our way towards a *paradox of definability without self-reference*. Let E_1, E_2, ... be an arbitrary finite or denumerable list of denoting expressions. Some of these expression may denote positive integers (i.e. 1, 2, 3, ...). We define the *max* function as follows:

> If n is the largest integer denoted by an expression on the list $E_1, E_2, \ldots$, then $\max(E_1, E_2, \ldots) = n$; if denumerably many distinct positive integers are denoted by expressions on the list (so that there is no largest positive integer denoted), $\max(E_1, E_2, \ldots) = \omega$, the first infinite ordinal; and if no expression on the list denotes a positive integer, $\max(E_1, E_2, \ldots) = 0$.

For example, max('London', ‘the only even prime’, 'the successor of 2', 'red') = 3; max('London', 'New York', 'L.A.') = 0; and max('one', 'three', 'five', ... 'thirty one', ...) = ω.

Now consider the following infinite list of denoting expressions:

D_1. $1+\max(D_2,...,D_n,...)$.

D_2. $1+\max(D_3,...,D_n,...)$.

.
.
.

D_n. $1+\max(D_{n+1},...,D_{n+i},...)$.

D_{n+1}.$1+\max(D_{n+2},...,D_{n+i},...)$.

.
.
.

The expression '$\max(D_1,D_2,...)$' roughly abbreviates 'the largest integer denoted by an expression on the list D_1, D_2,...' – roughly, because to be exact we need to include the conditions under which the outputs ω and 0 are produced. But whether we put things roughly or precisely, it's clear that in general an occurrence of '$\max(E_1,E_2,...)$' abbreviates a longer expression in which 'denotes' occurs.

Suppose towards a contradiction that for some arbitrary n, D_n denotes a positive integer, say p. Now D_n is given by:

D_n. $1+\max(D_{n+1},..., D_{n+i},...)$.

Since D_n denotes p, $\max(D_{n+1},..., D_{n+i},...) = p-1$. So there is an expression D_k among D_{n+1}, ..., D_{n+i}, ... which denotes p-1.[9] D_k is given by:

D_k. $1+\max(D_{k+1},..., D_{k+i},...)$.

Since D_k denotes p-1, $\max(D_{k+1},..., D_{k+i},...) = p-2$. So there is an expression D_l among D_{k+1}, ..., D_{k+i}, ... which denotes p-2. And so on. Continuing in this way (for p-3 more steps), we obtain an expression D_z which denotes 1. D_z is given by:

D_z. $1+\max(D_{z+1},..., D_{z+i},...)$.

Since D_z denotes 1, $\max(D_{z+1},..., D_{z+i},...) = 0$. By the definition of 'max', none of D_{z+1}, ..., D_{z+i}, ...

denote a positive integer. In particular,

(i) D_{z+1} does not denote a positive integer.

Now D_{z+1} is given by:

D_{z+1}. $1+\max(D_{z+2},...,D_{z+i},...)$.

Since none of D_{z+2}, ..., D_{z+i}, ... denote a positive integer, $\max(D_{z+2},...,D_{z+i},...) = 0$. But then D_{z+1} denotes 1+0. That is,

(ii) D_{z+1} denotes a positive integer, namely 1.

From (i) and (ii), we obtain a contradiction.

By our reductio argument, we have shown that no D_n denotes a positive integer, for any n. So for all n, $\max(D_n,...,D_{n+i},...) = 0$. In particular, then, $\max(D_2,...,D_n,...)=0$; $\max(D_3,...,D_n,...)=0$; and so on. But then D_1 denotes 1+0; D_2 denotes 1+0; and in general, D_n denotes 1+0. To sum up: no D_n denotes a positive integer, and every D_n denotes a positive integer (namely 1). We are landed in paradox.

Observe that this paradox does not display any self-reference: each denoting expression makes reference only to phrases further down the list. This suggests that an adequate solution to the paradoxes of definability must do more than avoid circularity or self-reference - the roots of these paradoxes go deeper.

This paradox of definability is obviously more complicated than the simple one generated by C. But the singularity treatment of it is essentially the same. The discourse here has the same overall structure: first, denoting phrases are displayed; second, we reason to the conclusion that they are pathological and fail to denote; and third, we reason past pathology, and find that the phrases do denote. When we first assess D_1, D_2, and the rest as pathological, we assess them by an unreflective denotation schema, analogous to the c_P-schema in the case of C. We find that all the D_n's are pathological, and fail to denote. And with the availability of this new information, the common ground shifts, and we move to a reflective context. We now assess D_1, D_2 ... by a new reflective schema, analogous to the r_C-schema. Just as we reason first that P does not denote and then conclude that it does, so we find first that D_1, and all the others, fail to denote, and then go on to conclude that they do. There is no contradiction here. Rather there is a shift in the denotation-schemas by which we assess D_1, D_2 ..., triggered by the shift to a reflective context.

The primary representation for D_1 is $\langle type(D_1),c_{D1},c_{D1}\rangle$ and the primary tree is given by:

$$\langle type(D_1),c_{D1},c_{D1}\rangle$$
/ \ …

$\langle type(D_2),c_{D2},c_{D1}\rangle$ $\langle type(D_3),c_{D3},c_{D1}\rangle$ …

/ \ … / \ …

$\langle type(D_3),c_{D3},c_{D2}\rangle$ $\langle type(D_4),c_{D4},c_{D2}\rangle$ … $\langle type(D_4),c_{D4},c_{D2}\rangle$ $\langle type(D_5),c_{D5},c_{D2}\rangle$ …

. . … . . … . . … . . …

. . . .

. . . .

It's easy to check that this primary tree is also D_1's pruned$_0$ tree, that every branch is infinite, and that no node repeats on any infinite branch – so every branch is a chain. It follows from our definitions that D_1 is a reflection-free pathological expression. And it's also straightforward to check that the same is true of D_n, for any n – all these expressions are reflection-free pathological expressions. Further, D_2, D_3, … are key singularities of the occurrence of 'denotes' in D_1; and by Symmetry, D_1 is a singularity of the occurrence of 'denotes' in D_1 as well. Similarly, D_3, D_4, …are key singularities of 'denotes' in D_2; and in general, D_{n+1}, D_{n+2}, …. are key singularities of ‘denotes’ in D_n. And by Symmetry, D_n is a singularity of the occurrence of ‘denotes’ in D_n.

Once we recognize D_1’s pathology, and we identify the key singularities, we can determine a value for D_1. We move to a context r_{D1} explicitly reflective with respect to D_1, and determine a value for D_1 on the basis of its pathologicality. The determination tree for D_1 is its pruned$_0$ tree (since D_1 is reflection-free). This tree maps a procedure for determining a value for D_1: we remove the key singularities D_2, D_3, … from the extension of ‘denotes’ in D_1, yielding the value 0 for max(D_2, D_3, …), since none of D_2, D_3, … denotes$_{D1}$ an integer. And since D_1 is a token of the expression ‘1+max(D_2,...,D_n,...)’ in which the occurrence of ‘denotes’ is represented by ‘denotes$_{cD1}$’, we obtain the value 1 for D_1. This is the reflectively established value for D_1. D_1 does not denote$_{cD1}$, but it does denote$_{rD1}$. And similarly for any D_n. So there is no contradiction: the paradox is resolved by the identification and exclusion of singularities.

Notice that we can extend our initial reasoning to obtain a case of *iteration*. Consider this apparently paradoxical continuation of our reasoning:

> We have now found that D_1 denotes 1. And similarly D_2 denotes 1, … Dn denotes 1, … And now we can infer that the largest integer denoted by D_2, D_3, … is 1. So max(D_2, D_3, …) = 1, and 1+max(D_2, D_3, …) = 2. But here is the expression D_1 again – so we have that D_1 denotes both 1 and 2. And we are landed back in paradox.

The context of our continuation is explicitly reflective with respect to all of D_1, D_2, … - in this context, the pathologicality of these expressions is explicitly recognized and a value for each of them has been reflectively established. Let D_ω be the class {D_1, D_2, …}, and let the context of our continuation be $r_{D\omega}$, to indicate that the context is reflective with respect to every member of D_ω. According to our contextual analysis, the reasoning goes wrong when it reintroduces D_1 to obtain a contradiction. Here is the analysis up to that point:

> We have now found that D_1 denotes$_{rD\omega}$ 1. And similarly D_2 denotes$_{rD\omega}$ 1, D_3 denotes$_{rD\omega}$ 1, … . And now we can infer that the largest integer denoted$_{rD\omega}$ by D_2, D_3, … is 1. So max(D_2,D_3, …) = 1, and 1+max(D_2, D_3, …) = 2.

This is fine so far. But in the last sentence, when we say that 1+max(D_2,D_3, …) = 1, the token of 'max(D_2,D_3, …)' abbreviates an expression in which the occurrence of 'denotes' is represented by 'denotes$_{rD\omega}$' – roughly, the expression 'the largest integer denoted$_{rD\omega}$ by D_2, D_3, …'. This token, call it $D_1^\dagger$, is distinct from D_1, since it contains an occurrence of denotes$_{rD\omega}$, not denotes$_{D1}$. So we go wrong when we identify D_1 with $D_1^\dagger$. D_1 denotes$_{rD1}$ 1, and $D_1^\dagger$ denotes$_{rD\omega}$ 2, and there is no contradiction.

More formally, the primary representation of $D_1^\dagger$ is given by <type(D_1), $r_{D\omega}$,$r_{D\omega}$>, and each branch of its pruned$_1$ tree terminates at the second branch:

<type(D_1),$r_{D\omega}$,$r_{D\omega}$>
/ | …
/ |
<type(D_2),c_{D2},$r_{D\omega}$> <type(D_3),c_{D3},$r_{D\omega}$> …

Here we assume an obvious extension to the definition of a *reflective δ-representation*, for δ ≥1. (The possibility of this extension was noted in endnote 4 of the previous chapter. In general, suppose r_{Cl} is a context explicitly reflective to a whole class of expressions Cl. Then a secondary representation of a δ-expression τ is a reflective δ-representation of τ if τ is a member of Cl and the third element of the representation is r_{Cl}.) The notation $r_{D\omega}$ indicates a context reflective to a class of expressions - in this case, the class D_ω - and we terminate any branch at the first occurrence of a secondary representation of a member of this class of the form <type(D_n),c_{Dn},$r_{D\omega}$>. This pruned$_1$ tree is also $D_1^\dagger$'s determination tree. The reflectively established value for each terminal node is 1. And from these values, we determine a value for

$D_1^\dagger$: we take the maximum of the values associated with the terminal nodes, add 1 to it, and obtain 2.

7.3.2 A Russell without circularity

Consider an infinite sequence of arbitrary 1-place predicates F_1, F_2, … F_k, … , and let S_k = {x| x=k or x is a non-empty finite extension of F_k or F_{k+1} or … F_{k+i} or … }. Now define the function ext* as follows:

> ext*(F_k,F_{k+1},…) = S_k if at least one of F_k, F_{k+1},… has a non-empty finite extension; otherwise, ext*(F_k,F_{k+1},…) = {Ø}, where Ø is the empty set.

For example, consider the following infinite sequence of 1-place predicates:

'integer between 1 and 5', 'natural number', 'NC Senator in 2013', '>0', '>1', … '>31', … .

Replacing the place-holders F_1, F_2,…, F_k, … by these predicates, we obtain

ext*(F_1,F_2,…F_k,…) = S_1 = {1, {2,3,4}, {Hagan, Burr}},

ext*(F_2,F_3,…F_k,…) = S_2 = {2, {Hagan, Burr}},

ext*(F_3,F_4,…F_k,…) = S_2 = {3, {Hagan, Burr}},

and ext*(F_4,F_5,…F_k,…) ={Ø}.

Now consider the following sequence of predicates:

P_1 member of ext*(P_2,P_3,…P_k,…)

P_2 member of ext*(P_3,P_4,…P_k,…)

.
.
.

P_k member of ext*(P_{k+1},P_{k+2},…)

.
.
.

Notice that the expression P_1 is, roughly, a short way of writing 'non-empty finite extension of P_2 or P_3 or … P_k or …' – roughly, because precision would require admitting 2 as a member of ext*(P_2,P_3,…P_k,…) and including the condition under which ext* yields {Ø}. But whether we put things roughly or precisely, the expression 'member of ext*(P_2,P_3,…P_k,…)' abbreviates a longer expression in which the term 'extension' occurs. The same goes for P_2, P_3, … P_k, … .

We will show that the sequence of predicates P_1, P_2, … P_k, … generates a paradox.[10] Let 'ext(P_k)' abbreviate 'the extension of P_k'.

<u>*Proposition*</u> ext(P_k)={Ø}, for arbitrary k.

<u>Proof</u> Suppose towards a contradiction that ext(P_k)≠{Ø}. Since ext(P_k)=ext*(P_{k+1},P_{k+2},…), it follows that

(i) for some m>k, P_m has a non-empty finite extension.

We can also show

(ii) ext(P_m) ≠{Ø}.

For suppose towards a contradiction that ext(P_m)={Ø}. Then, since ext(P_m)= ext*(P_{m+1},P_{m+2},…), no predicate among P_{m+1}, P_{m+2}, … has a non-empty finite extension. But consider P_{m+1}, namely 'member of ext*(P_{m+2},P_{m+3},…)'. Since no predicate among P_{m+2}, P_{m+3},… has a non-empty finite extension, ext*(P_{m+2},P_{m+3},…) = {Ø} = ext(P_{m+1}). But then P_{m+1} has a non-empty finite extension.[11] We have a contradiction, and this establishes (ii).

Given (i) and (ii), and since ext(P_m)= ext*(P_{m+1},P_{m+2},…), at least one of P_{m+1}, P_{m+2}, … has a non-empty finite extension. That is,

(iii) for some n>m, P_n has a non-empty finite extension.

We can also establish that

(iv) ext(P_n) ≠{Ø},

by reasoning exactly similar to that which established (ii).

Given (iii) and (iv), at least one of P_{n+1}, P_{n+2}, … has a non-empty finite extension – say, the predicate P_q. By reasoning exactly similar to that which established (ii), we can show that ext(P_q) ≠{Ø}. And so we obtain that at least one of P_{q+1}, P_{q+2}, … - say, P_r - has a non-empty finite extension. Again we can show that ext(P_r) ≠{Ø}. And so on: the reasoning may be repeated indefinitely. So there are denumerably many predicates P_n, P_q, P_r, … with non-empty finite extensions other than {Ø}, where n, q, r, … >m. These extensions are all distinct – each contains a positive integer peculiar to it (for example, n+1 is a member of ext(P_n) but not of ext(P_q) or ext(P_r) or …).[12] So there are denumerably many of these extensions - and they are all members of ext(P_m). So P_m has an infinite extension, contradicting (i). This completes the proof of the Proposition.

Since k is arbitrary, we have that for all k, ext(P_k)= {Ø}. In particular,

(a) ext(P_1)={Ø}.

Now P_1 is the predicate 'member of ext*(P_2,P_3,…P_k,…)', and all of P_2, P_3, … P_k, … have the same non-empty finite extension, namely {Ø}. So {Ø} is a member of ext*(P_2,P_3,…P_k,…), along with 2. So

(b) ext(P_1)={2,{Ø}}.

Since (a) and (b) yield a contradiction, we are landed in paradox.[13]

Like the paradox of definability, this paradox does not display any self-reference: each predicate makes reference only to predicates further down the list. Again, an adequate treatment of this paradox must go beyond considerations of self-reference and circularity.

The singularity theory resolves this paradox as follows. The primary representation of P_1 is $<\text{type}(P_1),c_{P1},c_{P1}>$, and the primary tree for P_1 is analogous to that of D_1:

$<\text{type}(P_1),c_{P1},c_{P1}>$

/ \ …

$<\text{type}(P_2),c_{P2},c_{P1}>$ $<\text{type}(P_3),c_{P3},c_{P1}>$ …

/ | … / \ …

$<\text{type}(P_3),c_{P3},c_{P2}>$ $<\text{type}(P_4),c_{P4},c_{P2}>$ … $<\text{type}(P_4),c_{P4},c_{P3}>$ $<\text{type}(P_5),c_{P4},c_{P3}>$ …

…

It's straightforward to check that according to the singularity theory, the expressions P_1 is a reflection-free pathological expression. This primary tree is also P_1's pruned$_0$ tree, and the tree indicates that P_2, P_3, … are key singularities of the occurrence of 'extension' in P_1. P_1's pruned$_0$ tree is also P_1's determination tree, and the tree maps out a procedure to establish reflectively a value for P_1. We exclude the key singularities from the extension of 'extension' in P_1. That is, none of P_2, P_3, … have an extension$_{cP1}$. *A fortiori*, none of P_2, P_3, … have finite extensions$_{cP1}$. Since P_1 is a token of 'member of ext*(P_2,P_3,…P_k,…)' in which the occurrence of 'extension' is represented by 'extension$_{cP1}$', it follows that the reflectively established value for P_1 is {Ø}. That is, P_1 does not have an extension$_{cP1}$, but its extension$_{rP1}$ is {Ø}.

Similarly, it's easy to check that P_2 is a reflection-free pathological expression, and P_3, P_4, … are key singularities of 'extension' in P_2. In general, P_{k+1}, P_{k+2}, …. are key singularities of ‘extension’ in P_k. And the reflectively established value for each of P_1, P_2, … is {Ø}.

Now consider the final stretch of the reasoning, where we derive a contradiction from (a) and (b). The contextual analysis of (a) is:

(a) $ext_{rP1}(P_1)$={Ø}.

That is, the reflectively established $extension_{rP1}$ of P_1 is {Ø}. In our original reasoning, we reach (b) as follows:

> Now P_1 is the predicate ‘member of ext*(P_2,P_3,…P_k,…)’, and all of P_2, P_3, … P_k, … have the same non-empty finite extension, namely {Ø}. So {Ø} is a member of ext*(P_2,P_3,…P_k,…), along with 2. So
> (b) ext(P_1)={2,{Ø}}.
> Since (a) and (b) yield a contradiction, we are landed in paradox.

In establishing (b) here, we rely on the reflectively established values of P_2, P_3, … P_k, … -- we are reasoning in a context which is reflective with respect to all of P_1, P_2, P_3, … P_k, … . Let P_ω be the class {P_1, P_2, P_3, … P_k, … }, and ‘rP_ω’ denote a context reflective with respect to this class of expressions.

Now it is correct that {Ø} and 2 are the members of ext*(P_2,P_3,…P_k,…) – but it is crucial to notice that the occurrence of ‘ext*(P_2,P_3,…P_k,…)’ in our final stretch of reasoning abbreviates an expression that contains an occurrence of ‘extension’ represented by ‘$extension_{rP\omega}$’. We obtain the value {2, {Ø}} for ‘ext*(P_2,P_3,…P_k,…)’ here from the reflectively established extensions of P_2, P_3, …P_k, … . So in our reasoning we produce a token $P_1^\dagger$ of the same type as P_1, but $P_1^\dagger$ contains an occurrence of ‘$extension_{rP\omega}$’ while P_1 contains an occurrence of ‘$extension_{cP1}$’.[14] This is a case of iteration, treated along the familiar lines. The $extension_{rP\omega}$ of P_1 is {Ø}, and the $extension_{rPk}$ of $P_1^\dagger$ is {2, {Ø}}.

It is easy to check (in analogy with the case of $D_1^\dagger$) that $P_1^\dagger$’s primary representation is

<type(P_1),$r_{P\omega}$,$r_{P\omega}$ >, and every branch of $P_1^{\dagger}$'s pruned$_1$ tree terminates at the second tier. Each terminal node is associated with the reflectively established value {Ø} for P_2, P_3, … P_k,… . This pruned$_1$ tree is also the determination tree for $P_1^{\dagger}$, and to determine a value for $P_1^{\dagger}$ we take the value {Ø} from the terminal nodes, and form the set {2, {Ø}}.

7.3.3 Truth paradoxes without circularity

Yablo's paradox runs as follows. Consider the following infinite sequence of sentences:

(S_1) for all k>1, s_k is untrue.

(S_2) for all k>2, s_k is untrue.

(S_3) for all k>3, s_k is untrue.

.
.
.

Each sentence says that every subsequent sentence is untrue.

We can now reason as follows. Suppose towards a contradiction that some sentence S_n is true. Then, given what S_n says, it follows that

(i) for all k>n, S_k is not true.

And from (i) it follows both that

(ii) S_{n+1} is not true

and that

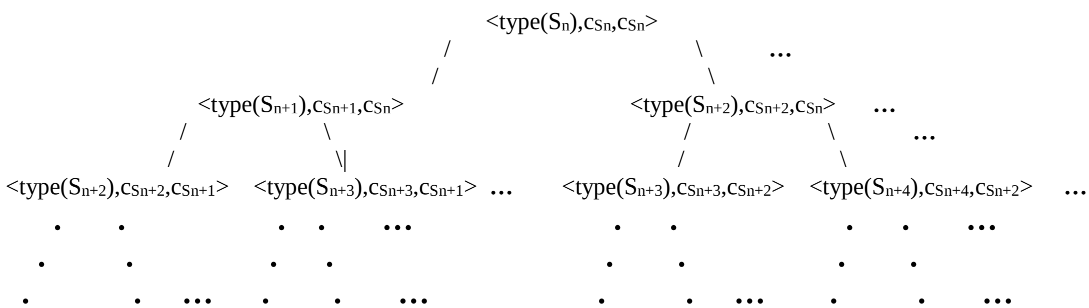
(iii) for all k>n+1, S_k is not true,

But S_{n+1} says just what (iii) says, and so

(iv) S_{n+1} is true.

We have a contradiction at lines (ii) and (iv), and we conclude that every sentence S_n is not true. But then every sentence subsequent to a given S_n is not true, just as S_n says. So S_n is true. And we are landed in contradiction.

The singularity treatment of Yablo's paradox runs as follows. The primary representation of S_n is $\langle \text{type}(S_n),c_{S_n},c_{S_n}\rangle$, and its pruned_0 tree is just its primary tree:



where in general c_{S_k} is the context of utterance of S_k. Since every branch of this pruned_0 tree is a chain, S_n is a pathological reflection-free expression. We can check that S_n, S_{n+1}, … form a reflection-free symmetrical network. And so by Symmetry all of S_1, S_2, …, S_n, … are pathological. This conclusion, that S_1, S_2, …, S_n, … are pathological, corresponds to the contradiction we reach by the end of the reasoning of Yablo's paradox. We can summarize the reasoning in terms of our contextual analysis. Recall from Chapter 9 that the occurrences of the semantic term 't' in the members of a reflection-free symmetric network are coextensive; so in particular, the occurrences of 'true' in S_1, S_2, … S_n, … are coextensive. So let each of these occurrences be represented by 'true_{cS}', where c_S is the context shared by the symmetric network of sentences S_1, S_2, … , a context that is unreflective with respect to these sentences. Then we can represent in overview the reasoning that constitutesYablo's paradox as follows: we start out with the assumption that S_n is true$_{cS}$, and reach a contradiction – so S_n is not true$_{cS}$. But this goes for any S_n – so every S_n is untrue$_{cS}$. And so each S_n is true$_{cS}$, given what it says. So the attempt to evaluate any S_n by the c_S-schema leads to contradiction.

The paradox is resolved by the identification and exclusion of singularities. We can check that all the members of S_n's determination set - namely, S_{n+1}, S_{n+2}, ... S_{n+k}, ... - are key singularities of the occurrence of 'true' in S_n. Once the key singularities of 'true' in S_n are removed, the contradiction is removed.

However, it is a virtue of the singularity theory that we can take things further. We can regard Yablo's paradox as just the first stage in a stretch of strengthened liar reasoning. Things need not end there – we can reason past pathology. Once we recognize that S_1, S_2, … S_n, … are pathological, we can move to a context r_{Sn} reflective with respect to S_n for any n. The determination tree for S_n is just S_n's pruned$_0$ tree, and to determine a value for S_n, we eliminate each of the key singularities – as indicated at the second tier of the tree - from the extension of 'true' in S_n. That is, none of S_{n+1}, S_{n+2}, ... are true$_S$. But when we look at what S_n says - that none of S_2, S_3, ... are true$_S$ - we find that S_n is true$_{rSn}$. This is a reflectively established value for S_n. In the familiar way, we can distinguish two contexts. There is the unreflective context c_S in which we reason to the conclusion that each of S_1, S_2, ... are pathological. And there is the subsequent reflective context in which we reason to the conclusion that S_n has a determinate truth value, on the basis of what S_n says and the identification and exclusion of the key singularities of 'true' in S_n.

***

The paradoxes we have considered in 7.3 – involving denotation, extension and truth –

display no self-reference or circularity, and yet they each generate contradictions. So they are of special interest. But as we noted at the outset of 7.3.1, regarding denotation, neither circularity nor contradiction is needed for pathology. For another example, one can easily construct variants of Yablo's paradox that are pathological but do not yield a contradiction. Consider the sequence:

(S_1) for all $k>1$, s_k is true.

(S_2) for all $k>2$, s_k is true.

(S_3) for all $k>3$, s_k is true.

.
.
.

Suppose first that a given S_n is true. Then every subsequent sentence is true. It's easy to see that we can consistently assign truth to every sentence in the sequence. Suppose second that a given S_n is not true. Then at least one subsequent sentence is not true, rendering all previous sentences untrue. And under these circumstances we can consistently assign untruth to every sentence in the sequence. The problem here is not that we are landed in contradiction - rather, it is the problem that we have no reason to give S_n one truth value rather than another. We can assign consistently assign truth to each S_n, and we can consistently assign untruth to each S_n. (Compare the Truth-Teller.) We would expect any adequate account of semantic paradox to handle this variant of Yablo's paradox too. It is straightforward to check that according to the singularity solution each sentence in the sequence is pathological, and is false upon reflection - each says it is true in the initial unreflective context, and it isn't. (Similarly, as we've seen, the Truth-Teller is false upon reflection.)

Notes to Chapter 7

1. So the Truth-Teller is dealt with straightforwardly. In contrast, the Truth-Teller poses special problems for Field's theory of truth, as we'll see in Chapter 8.

2. Note that the initial presentation of the Curry use quotation marks. But according to my contextual approach, we have to pay careful attention to the distinction between types and tokens – the bearers of truth are sentence tokens (or sentence types paired with contexts). On my account of the Curry, or any of the paradoxes, we refer to particular token expressions, and quote names cannot serve this purpose. Compare note 72, Chapter 5.

3. The singularity theory presented in Chapter 6 is at a high enough level of abstraction to encompass not only truth, but the other semantic concepts too. In Simmons 1993, the focus is more narrowly on truth, and a fully detailed formal account of the evaluation of the Curry sentence is available there, pp.148-9.

4. For discussion, see Beall 2008, section 3.

5. Armour-Garb and Woodbridge discuss the Open Pair paradox in their 2005, 2006, 2007 and 2008.

6. Kripke 1975, in Martin 1984, pp.59-60.

7. Again, the focus of Simmons 1993 is on truth, and a fully detailed account of the evaluation of (1) and (2) in the present circumstances is available there, pp.142-3.

8. According to Scharp, contextual accounts of the Nixon-Dean case would require the pathological character of the utterances to be part of the common ground, information explicitly available to the participants – see Scharp 2013, 3.2.4. As we saw in 3.2**,** the singularity theory makes no such requirement. Our semantical predicates are sensitive to the reflective status of their contexts of use, and reflective status can be determined either explicitly or non-explicitly. In the latter case, the reflective status of the context is determined by the semantic network generated by the given expression, and the specifics of the network may not be available to the participants. In particular, the semantic network generated by Nixon's utterance (1) determines that (1)'s context is not reflective with respect to Dean's (2), quite independently of whether Nixon knows that (1), or (2), are pathological. The semantic network shows that (1) does not stand above (2)'s pathology – (1) and (2) are pathologically looped, and so each is identified as a singularity of the occurrences of 'true' in (1) and (2).

9. There can be only one such expression. Suppose, towards a contradiction that D_m and D_n each denote p-1, where we may assume, without loss of generality, that m<n. Then D_m is given by:
D_m. 1+max(D_{m+1},...D_n,...).

Since D_m denotes p-1, max(D_{m+1},...D_n,...)=p-2. But since D_n denotes p-1, max(D_{m+1},...D_n,...)$\geq$ p-1. So p-2$\geq$p-1, and we have a contradiction.

10. This paradox about extensions is somewhat more complicated than the denotation paradox. I am doubtful though that a simpler version can be found. One might wonder if we could drop the disjunct x=k in the definition of S_k. But we cannot - see note 12 below. And neither can we replace {Ø} by Ø in the definition of ext* - see note 11.

11. This is why {Ø} rather than Ø appears the definition of ext*: {Ø} is a non-empty (finite) set, but Ø is not.

12. This is why the disjunct x=k appears in the definition of S_k.

13. This genuine paradox should be sharply distinguished from simpler pseudo-paradoxes. For example, consider an omega-length string of predicates P_k of natural numbers as follows:

P_k is the 1-place predicate 'x=1' if 1 is not in the extension of any later P_n

P_k is the 1-place predicate 'x≠1' if 1 is in the extension of some later P_n.

If we suppose that 1 is in the extension of some Pk, we reach a contradiction. So 1 is not in the extension of any Pk. So each Pk is 'x≠1'. So 1 must be in the extension of some P_n with n>k. Contradiction.

Now this 'paradox' is easily resolved: we simply conclude that there is no string of predicates that satisfies the given conditions. None of the predicates can be 'x=1', but if none of them are, then some of them are. (Just as there can be no Barber, so there can be no such string of predicates.) This pseudo-paradox poses no problem for the notion of extension. The same cannot be said of the paradox I have presented, where a sequence of predicates is explicitly laid out. There is no issue about what these predicates are. They are given, just as the denoting phrases D_1, D_2 … are given (and just as the sentences are given in Yablo's non-circular version of the Liar, in Yablo 1993). So we can ask after their extensions (just as we can ask after the denotations of D_1, D_2, …). And in doing so we are led to paradox. The notion of extension gives rise to a paradox – a paradox that does not turn on self-reference or circularity.

14. Notice that we have to be careful about the use of quotes (compare note 2 above, and note 72, Chapter 5). In the final stretch of reasoning, we say: "Now P_1 is the predicate 'member of ext*(P_2,P_3,…P_k,…)'". Failure to be sensitive to the type-token distinction here can encourage the appearance of paradox. What we can say is that P_1 is a token of the type 'member of ext*(P_2,P_3,…P_k,…)'. Another token of the very same type appears in the next sentence. But this *is* another token, and the occurrence of 'extension' that occurs in the expression it abbreviates is differently represented from the occurrence associated with P_1.