8.6. Arguments involving descriptive reference

8.6.0. Overview

When definite descriptions are given Russell's analysis, their properties follow from the properties of the logical constants used in their analysis, but the description operator requires special treatment.

Glen Helman 01 Aug 2013

8.6.1. The role of definite descriptions in entailment

If Russell's analysis of definite descriptions is accepted, their logical properties follow from those of the logical constants used in the analysis; but the description operator is a new symbol, and studying its logical properties requires stating new principles for it. We looked at the logical properties of the description operator informally in 8.4.3. Now we will look at a way of stating a principle of entailment that captures these properties.

First, we must first find a place for the description operator in our semantic scheme. All our logical constants so far—whether the connectives, the quantifiers, or the identity predicate—have been ways of producing compound formulas. The description operator, on the other hand, yields a compound term when it is applied to a predicate. This means that the extension of the operator I will be a function from the extensions of one-place predicates to reference values. We can represent the extension of a one-place predicate by the set of reference values of which it is true, so the extension of the description operator can be seen as a function which takes a set of reference values as input and yields a single reference value as output.

According to the account of definite descriptions we are considering, a term Ix ρx formed using the description operator refers to the single value in the extension of ρ if there is just one value, and otherwise its reference value is the Nil. This means that the extension of the description operator is not settled until we identify the Nil as a specific value in the referential range. This identification must be considered a further component of a structure, a respect in which two structures may differ. So when we make the description operator a part of our language, we require that a structure distinguish a member of the referential range as the Nil. This will serve as the reference value of the constant individual term ∗ introduced in 8.4.3. Then, to find the semantic value given to Ix ρx by a structure, we find the extension the structure gives to the predicate ρ. If the extension of ρ has just one member, that reference value will be the extension of Ix ρx; otherwise, the extension of Ix ρx is the value the structure assigns to ∗.

A specification made regarding structures and the interpretation of logical vocabulary will typically result in some logical law. For example, the requirement that the referential range serve both as a source of extensions for terms and as the domain of unrestricted universals gives us the principle of universal instantiation. And even the simple requirement that a referential range be non-empty yields the law $\forall x\ \theta x \vDash \exists x\ \theta x$, which assures us that universal predicates are exemplified. In the case of our specifications for definite descriptions and the Nil, we get a principle that identifies a certain sentence as a tautology.

LAW FOR DESCRIPTIONS.

⊨ (∃z: ρz ∧ (∀y: ρy) z=y) ℩xρx = z
∨ ((∀x: ρx)(∃y: ρy) ¬ x=y ∧ ℩xρx = ∗) (for any predicate ρ)

This tautology is a disjunction whose two components express the two alternatives for the reference value of a definite description. Let us see how that works in a little more detail.

The existential quantifier in the first disjunct should be familiar as one way of writing the quantifier that Russell used to analyze definite descriptions. The whole first disjunct might be read as Something such that (ρ fits it and it is all that ρ fits) is such that (the thing that ρ fits is it) or, a little more idiomatically, as The thing that ρ fits is something that ρ fits uniquely.

The second disjunct of the sentence is a conjunction whose first conjunct says Anything that ρ fits is such that something ρ fits is different from it. This is a compact but somewhat roundabout way of saying that the extension of ρ does not have exactly one member—i.e., if we can find anything in it, we can find something else in it, too. The second conjunct of this part of the sentence can be read as The thing that ρ fits is the Nil.

Putting this all together, the law tells us that the following is a tautology:

Either (i) ℩x ρx refers to something that ρ fits uniquely, or (ii)
ρ does not fit exactly one thing and ℩x ρx refers to the Nil

The first disjunct specifies the reference of the definite description when this is determined by the description, and the second disjunct specifies the reference when the description does not succeed in determining it.

In 8.4.3 the content of an analysis using the description operator was expressed using a similar disjunction. On that account, a sentence θ(℩x ρx) says that either (i) ρ is true of exactly one thing and (∃x: ρx) θx is true or (ii) ρ is not true of exactly one thing and θ∗. Given the law for descriptions, the properties of identity will tell us that

θ(℩x ρx) ≃ ((∃x: ρx) (∀y: ρy) x = y ∧ (∃x: ρx) θx)
∨ ((∀x: ρx) (∃y: ρy) ¬ x = y ∧ θ∗)

and the right-hand side is a more formal version of the disjunction used in 8.4.3.

8.6.2. Derivations for the description operator

Although, in stating the tautologousness of a single long sentence, the law for the description operator takes a somewhat different form than those we considered for other logical constants, the real novelty in handling this constant lies in the fact that it is used to form terms rather than sentences. This means that what we must account for is not the role of a premise or conclusion. Instead, we need to account for what a definite description refers to.

The law for the description operator provides a way to draw conclusions about what a definite description refers to. We will implement this law in a rule that amounts to a couple steps in the exploitation of the sentence the law asserts to be a tautology. In particular, our rule will lead us directly to what we would get as the result of using a proof by cases to exploit the disjunctive law (restated using unrestricted quantifiers) and then using proof by choice and extraction for its existential first disjunct. The remaining non-atomic sentences in the law are universals so we cannot expect to go further in a single step. We will call this rule *Securing a Description* (SD).

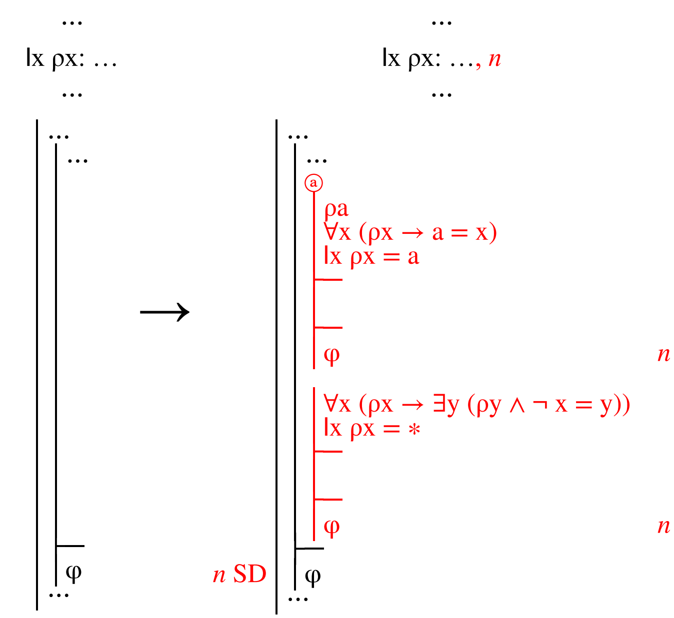


Fig. 8.6.2-1. Developing a derivation at stage *n* by securing a definite description; the independent term a is new to the derivation.

There are really no preconditions for the use of this rule, but it is relevant only when the definite description in question actually appears in the gap being developed. The description is displayed above the derivation (perhaps among a list of other definite descriptions) and the stage number of the development is listed after it to show that it has been handled—we will say *secured*—at that stage in developing some gap. The description may need to be secured in a number of different gaps at different stages, so this stage is perhaps only the latest of a long list.

The term secure was used in 7.8.1 in connection with the rule ST, which was intended to provide a way to locate finite structures when the normal development of a gap would introduce ever more complex compound terms. Our aim in the rule SD is different but the consequences are similar. When a definite description is secured, it will be in the same alias set as some simple term, either the independent term introduced in the first gap or the term ∗. However, SD is not designed to search for finite structures, and we are as interested in the other assumptions introduced in each of the two gaps as in the equations that actually secure the description.

The occurrence of the definite description operator in an argument forces us to consider ∗ among the terms which will be grouped into alias sets. This is not merely because this term will be introduced into one of the gaps that is the result of applying SD. The semantics of the description operator require that range of reference values contain a distinguished nil value. This value must be included among the reference values of the terms for which we exploit universals if we are to insure that the universal is true. And that means we will need to exploit a universal for ∗ (or a co-alias) before a gap reaches a dead end. This is true even if ∗ does not actually appear in the gap (in which case it will be its only co-alias). Thus the terms to be considered when forming alias sets are not merely the terms appearing the resources and goals of the gap and its ancestors but also ∗ if any definite description appears among the terms we need to consider. Although it is possible for new definite descriptions to appear in the course of a derivation, this will happen only if one appears in the initial argument, so the requirement for including ∗ is that it must be counted among the terms whenever a definite description appears in the argument whose validity we are considering.

As an example of the use of SD, here is a derivation showing that if have the premise There was at most one winner, we can conclude The winner won if anything did.

Ix Wx: 3

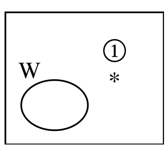
	¬ ∃x ∃y (¬ y = x ∧ (Wx ∧ Wy))	(14)
	ⓐ	
	Wa	(6), (10)
	ⓑ	
	Wb	(4)
	∀x (Wx → b = x)	
	Ix Wx = b	a, b—IxWx, ∗
	●	
4 QED=	W(Ix Wx)	3
	∀x (Wx → ∃y (Wy ∧ ¬ x = y))	a:5
	Ix Wx = ∗	a, c, IxWx—∗
5 UI	Wa → ∃y (Wy ∧ ¬ a = y)	6
6 MPP	∃y (Wy ∧ ¬ a = y)	7
	ⓒ	
	Wc ∧ ¬ a = c	8
8 Ext	Wc	(10)
8 Ext	¬ a = c	(11)
	¬ W(Ix Wx)	
10 Adj	Wc ∧ Wa	X, (11)
11 Adj	¬ a = c ∧ (Wc ∧ Wa)	X, (12)
12 EG	∃y (¬ y = c ∧ (Wc ∧ Wy))	X, (13)
13 EG	∃x ∃y (¬ y = x ∧ (Wx ∧ Wy))	X, (14)
	●	
14 Nc	⊥	9
9 IP	W(Ix Wx)	7
7 PCh	W(Ix Wx)	3
3 SD	W(Ix Wx)	2
2 CP	Wa → W(Ix Wx))	1
1 UG	∀y (Wy → W(Ix Wx))	

Notice that the list of alias sets in the first gap includes ∗ even though that term does not appear in either resources or goals of that gap because a definite description does appear in the initial conclusion.

Notice also that both the premise and the hedge if anything did in the conclusion played a role in closing the second gap in the derivation above. Since both are required to insure the existence and uniqueness of a winner, it is to be expected that the absence of either would keep us from ruling out the possibility that the definite description is undefined (which is the possibility explored by the second gap).

It may seem odd that The winner won is not a tautology. But on both of the accounts of definite descriptions that we have considered, it entails Something won, and that is not a tautology. It follows that The winner didn't win is not absurd if it is contradictory to The winner won, and a derivation showing this when the sentence is interpreted using the description operator provides another example of the use of SD.

```
      ℩x Wx: 1
        │ ¬ W(℩x Wx)                          (2)
        ├─
        │ ⓐ
        │ │ Wa                                (2)
        │ │ ∀y (Wx → a = x)
        │ │ ℩x Wx = a                         (℩x Wx)—a
        │ ├─
        │ │ ●
        │ ├─
2 Nc=   │ │ ⊥
        │
        │ │ ∀y (Wx → ∃y (Wy ∧ ¬ x = y))       ∗:3
        │ │ ℩x Wx = ∗                         (℩x Wx)—∗
        │ ├─
3 UI    │ │ W∗ → ∃y (Wy ∧ ¬ ∗ = y)            4
        │ │ │ │ ¬ W∗
        │ │ │ ├─
        │ │ │ │ ○                             ¬ W(℩x Wx), ℩x Wx = ∗,
        │ │ │ │                                  ¬ W∗ ⊭ ⊥
        │ │ │ ├─
        │ │ │ │ ⊥                             5
        │ │ ├─
5 IP    │ │ │ W∗                              4
        │ │
        │ │ │ ∃y (Wy ∧ ¬ ∗ = y)
        │ │ ├─
        │ │ │ (unfinished)
        │ │ ├─
        │ │ │ ⊥                               4
        │ ├─
4 RC    │ │ ⊥                                 1
        ├─
1 SD    │ ⊥
```



The definite description ℩x Wx does not appear in the diagram of the counterexample because, as a compound expression, its value is determined by the values shown there. In particular, the fact that the extension of W is empty insures that ℩x Wx has the same reference value as ∗, and that would be true even if the referential range contained more than this value.

The sentence The winner didn't win is contingent also on Russell's analysis provided we interpret it as the denial of The winner won. For the latter sentence will be contingent according to Russell's analysis since it is true on that analysis if and only if there is exactly one winner. However, on Russell's analysis, an interpretation giving the winner widest scope—that is, an interpretation of the sentence as The winner is such that (he or she didn't win)—is absurd since it implies Some winner didn't win and thus that something has both the property of winning and the property of not winning. This is another consequence of the ambiguity that can arise with definite descriptions on Russell's account. The sentence The winner won is definitely contingent on his way of analyzing it, but The winner didn't win may be either contingent or absurd, depending on whether the negation or definite description is understood to have wider scope.

Glen Helman 01 Aug 2013

8.6.3. Consequences for adequacy

Our stipulations about the interpretation of definite descriptions insure that any interpretation of the vocabulary in a description ℩x ρx will be (or can be extended so that it is) a counterexample to one of the two gaps that result from SD—that's why there is no precondition for the application of this rule—so SD is strict and its addition will not disturb the soundness of our system. SD is also clearly safe since the new gaps it introduces differ from their parent only by having added resources. But the argument we had used to establish the completeness of the system of derivations—in particular, the argument used in 7.7.4 to show that any fully developing gap has a counterexample lurking throughout—will no longer apply since this argument assumed that the reference values of all terms could be settled without considering the extensions of predicates, something that is not true in the case of definite descriptions.

We will not consider ways of reformulating that argument for a system including SD. Instead we will consider the completeness of a system of derivations for definite descriptions that employs not only SD but also certain uses of the rule LFR. The stipulations we have made concerning the interpretation of the description operator can be imposed on a structure simply by requiring that it make true every sentence of the form:

$$\forall w_1 \ldots \forall w_n \,(\,(\exists z{:}\ \rho z \land (\forall y{:}\ \rho y)\ z = y)\ ℩x\ \rho x = z \\ \lor ((\forall x{:}\ \rho x)\ (\exists y{:}\ \rho y)\ \neg\, x = y \land ℩x\ \rho x = *)\,)$$

where we follow the form of the law for descriptions but apply a quantifier $\forall w_i$ for each variable w_i that appears unbound in ρ. We will call this sentence a *meaning postulate* for the description ℩x ρx. Making all these meaning postulates true comes to the same thing as making true all instances of that law for a language expanded by the range of the structure. When assessing the validity of a particular argument, all that is relevant is the interpretation of the definite descriptions actually appearing in the argument (provided we take this to include descriptions containing variables that are not bound within the description). And the correct intepretation of these descriptions can be insured by the truth of the meaning postulates for them. That is, if Δ includes the meaning postulate for each description in an argument Γ / φ, this argument is valid given the interpretation of the description operator if and only if the argument Γ, Δ / φ is valid even without stipulating the interpretation of definite descriptions—i.e., even if we treat them as unanalyzed individual terms.

Now, any question of validity can be reduced to a question of the validity of a *reductio* argument, so let us limit consideration to such arguments. Given an argument Γ / ⊥, let δ be the conjunction of the meaning postulates for all descriptions appearing in the members of Γ. Now suppose that Γ / ⊥ is valid when we fix the interpretation of definite descriptions. We have seen that Γ, δ / ⊥ will be valid without fixing this interpretation. Therefore, a derivation for Γ, δ / ⊥ will close using only the basic system of previous chapters, so it will certainly close if we add the rules SD and LFR. And the rule SD will enable us to show that the meaning postulate for any description is a tautology, so it will certainly enable us to show the validity of Γ / δ. Finally, the rule LFR lets us establish the validity of Γ / ⊥ if we can show both Γ ⊨ δ and Γ, δ ⊨ ⊥. In short, the system of derivations with SD and LFR is complete because SD enables us to establish any meaning postulate, and we can establish the validity of all arguments involving descriptions when we add their meaning postulates as further premises.

Since it introduces a new independent term, the rule SD introduces a new way that gaps can be prevented from reaching a dead end. It can be modified to search for finite structures in the way we have done for other rules using independent terms, and named, following the same pattern as with those rules, as *Securing a Description Supplemented* (SD+).

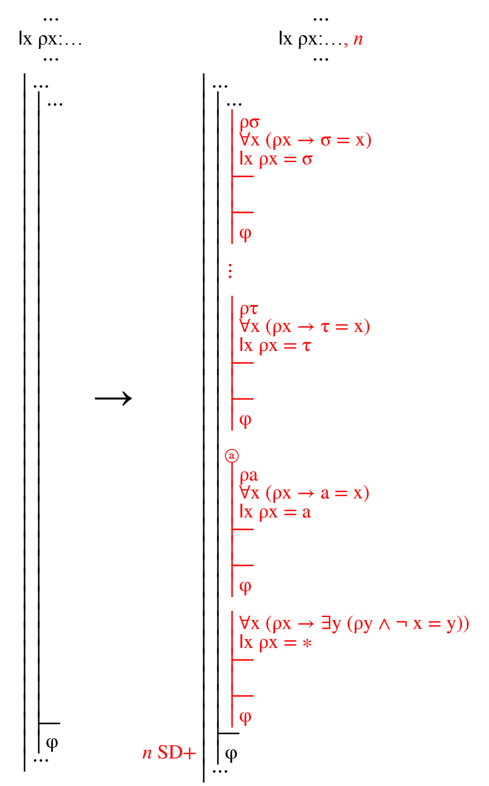


Fig. 8.6.3-1. Developing a derivation at stage *n* by securing a definite description; the independent term a is new to the derivation and the terms σ, …, τ include at least one from each current alias set for the gap.

When we use this rule, we consider the possibility that one of the already existing alias sets provides names of an object that uniquely satisfies the description.

Notice that one of these alias sets will be the one including ∗. And that is to be expected since there are two different ways in which the Nil might end up as the reference value of a definite description. This will happen not only when the description fails to be uniquely satisfied but also when the Nil is the one value satisfying it uniquely. Indeed, the reference of any term τ will uniquely satisfy the predicate [_ = τ], so whether or not [_ is a C] is not uniquely satisfied [_ = the thing that is a C] will be—though, of course, perhaps only by the Nil.

8.6.s. Summary

1 In order to assign a meaning to the description operator with respect to a referential range, a reference value must be singled out as the Nil. This serves as the reference value of the constant ∗ and as the reference value of the description Ix ρx when the extension of ρ is empty or has more than one member. Then the law for descriptions asserts that either (i) Ix ρx is something that is the sole thing ρ is true of or (ii) ρ is not true of exactly one thing and Ix ρx has the Nil as its reference value.

2 A definite description is not a sentence, so it is handled in derivations not by exploiting it or planning for it as a goal but by securing it—that is, by insuring that its reference is settled in the way required by the law for descriptions. The rule for doing this is Securing a Description (SD).

3 This rule is enough to enable us to establish meaning postulates, which state that definite descriptions are interpreted as we intend. Although the argument used for completeness of the system of derivations no longer applies, it is easy to see that the system is complete if we allow the rule LFR to be used to introduce meaning postulates as lemmas. The rule SD introduces a new term, so to search for finite counterexamples, we need an alternative form, Securing a Description Supplemented (SD+).

8.6.x. Exercise questions

Analyze each of the following first using Russell's approach to definite descriptions and then again using the description operator. Use derivations to check each form of the argument for validity.

1. The winner was an amateur
   ―――――――――――――
   An amateur was a winner

2. An amateur was a winner
   There was at most one winner
   ―――――――――――――
   The winner was an amateur

The exercise machine doesn't incorporate rules for the description operator; but you can use it for the logical forms derived from Russell's analysis.

Glen Helman 01 Aug 2013

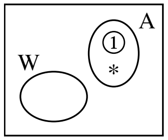
8.6.xa. Exercise answers

1.

	∃x ((Wx ∧ ∀y (¬ y = x → ¬ Wy)) ∧ Ax)	1
	ⓐ (Wa ∧ ∀y (¬ y = a → ¬ Wy)) ∧ Aa	2
2 Ext	Wa ∧ ∀y (¬ y = a → ¬ Wy)	3
2 Ext	Aa	(4)
3 Ext	Wa	(4)
3 Ext	∀y (¬ y = a → ¬ Wy)	
4 Adj	Aa ∧ Wa	X, (5)
5 EG	∃x (Ax ∧ Wx)	X, (6)
	●	
6 QED	∃x (Ax ∧ Wx)	1
1 PCh	∃x (Ax ∧ Wx)	

℩x Wx: 3

	A(℩x Wx)	(3)
	∀x ¬ (Ax ∧ Wx)	℩x Wx:2
2 UI	¬ (A(℩x Wx) ∧ W(℩x Wx))	3
3 MPT	¬ W(℩x Wx)	(5)
	ⓐ Wa	(5)
	∀x (Wx → a = x)	
	℩x Wx = a	℩xWx—a, ∗
	●	
5 Nc=	⊥	4
	∀x (Wx → ∃y (Wy ∧ ¬ x = y))	∗:6
	℩x Wx = ∗	℩xWx—∗
6 UI	W∗ → ∃y (Wy ∧ ¬ ∗ = y))	7
	¬ W∗	
	○	A(℩x Wx),¬ W(℩x Wx), ¬ W∗,(℩xWx)=∗ ⊭ ⊥
	⊥	8
8 IP	W∗	7
	∃y (Wy ∧ ¬ ∗ = y)	
	(unfinished)	
	⊥	7
7 RC	⊥	4
4 SD	⊥	1
1 NcP	∃x (Ax ∧ Wx)	



2.

	∃x (Ax ∧ Wx)	1
	¬ ∃x ∃y (¬ y = x ∧ (Wx ∧ Wy))	(15)
	ⓐ Aa ∧ Wa	2
2 Ext	Aa	(5)
2 Ext	Wa	(6), (11)
	∀x ¬ ((Wx ∧ ∀y (¬ y = x → ¬ Wy)) ∧ Ax)	a:4
4 UI	¬ ((Wa ∧ ∀y (¬ y = a → ¬ Wy)) ∧ Aa)	5
5 MPT	¬ (Wa ∧ ∀y (¬ y = a → ¬ Wy))	6
6 MPT	¬ ∀y (¬ y = a → ¬ Wy)	7
	ⓑ ¬ b = a	(12)
	Wb	(11)
11 Adj	Wa ∧ Wb	X, (12)
12 Adj	¬ b = a ∧ (Wa ∧ Wb)	X, (13)
13 EG	∃y (¬ y = a ∧ (Wa ∧ Wy))	X, (14)
14 EG	∃x ∃y (¬ y = x ∧ (Wx ∧ Wy))	X, (15)
	●	
15 Nc	⊥	10
10 RAA	¬ Wb	9
9 CP	¬ b = a → ¬ Wb	8
8 UG	∀y (¬ y = a → ¬ Wy)	7
7 CR	⊥	3
3 NcP	∃x ((Wx ∧ ∀y (¬ y = x → ¬ Wy)) ∧ Ax)	1
1 PCh	∃x ((Wx ∧ ∀y (¬ y = x → ¬ Wy)) ∧ Ax)	

Ix Wx: 2

	∃x (Ax ∧ Wx)	1
	¬ ∃x ∃y (¬ y = x ∧ (Wx ∧ Wy))	(16)
	ⓐ Aa ∧ Wa	2
2 Ext	Aa	(6)
2 Ext	Wa	(5), (8), (12)
	ⓑ Wb	
	∀y (Wy → b = y)	a:4
	Ix Wx = b	a, b—IxWx, ∗
4 UI	Wa → b = a	5
5 MPP	b = a	a—b—IxWx, ∗
	●	
6 QED=	A(Ix Wx)	3
	∀x (Wx → ∃y (Wy ∧ ¬ x = y))	a:7
	Ix Wx = ∗	a, c, IxWx—∗
7 UI	Wa → ∃y (Wy ∧ ¬ a = y)	8
8 MPP	∃y (Wy ∧ ¬ a = y)	9
	ⓒ Wc ∧ ¬ a = c	10
10 Ext	Wc	(12)
10 Ext	¬ a = c	(13)
	¬ A(Ix Wx)	
12 Adj	Wc ∧ Wa	X, (13)
13 Adj	¬ a = c ∧ (Wc ∧ Wa)	X, (14)
14 EG	∃y (¬ y = c ∧ (Wc ∧ Wy))	X, (15)
15 EG	∃x ∃y (¬ y = x ∧ (Wx ∧ Wy))	X, (16)
	●	
16 Nc	⊥	11
11 IP	A(Ix Wx)	9
9 PCh	A(Ix Wx)	3
3 SD	A(Ix Wx)	1
1 PCh	A(Ix Wx)	

Glen Helman 01 Aug 2013