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 25 Aug 2005

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. Studying the logical properties of the description operator is part of what must be done to decide on the correct analysis of English, and these properties were already discussed informally in 8.4.2.

To give a more explicit account of them, 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 ℩ 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 sets of reference values as input an yields single reference values as output.

We have required that a term ℩x ρx formed using the description operator refer to the single value in the extension of ρ if there is just one value and that it refer to the nil value otherwise. This means that the extension of the description operator is not settled until we identify the nil value 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 value. This will serve as the reference value of the constant individual term ∗ introduced in 8.4.2. Then, to find the semantic value given to ℩x 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 ℩x ρx; otherwise, the extension of ℩x 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 $\forall x\ \theta x \Rightarrow \exists x\ \theta x$, which assures us that universal predicates are exemplified. In the case of our specifications for definite descriptions and the nil value, we get what we will refer to as the ***law for descriptions***. In the case of a definite description $\iota x\ \rho x$ with no free variables, this takes the form:

$$\Rightarrow (\exists z\colon \rho z \wedge (\forall y\colon \rho y)\ z{=}y)\ \iota x\ \rho x = z \vee ((\forall x\colon \rho x)(\exists y\colon \rho y)\ \neg\, x{=}y \wedge \iota x\ \rho x = *).$$

The existential quantifier in the first disjunct should be familiar from Russell's analysis of 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* ρ *fits is it* or, a little more idiomatically, as *The thing* ρ *fits is something that is all that* ρ *fits*. The second disjunct of the law 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. Finally, the last component of the law can be read as *The thing* ρ *fits is the nil*. Putting this all together, the law amounts to the following:

> *Either (i)* $\iota x\ \rho x$ *refers to something that is all that* ρ *fits, or (ii)* ρ *does not fit exactly one thing and* $\iota x\ \rho x$ *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 8.4.2 the content of an analysis using the description operator was expressed using a similar disjunction. On that account, a sentence $\theta(\iota x\ \rho x)$ says that either ρ is true of exactly one thing and $(\exists x\colon \rho x)\ \theta x$ or ρ is not true of exactly one thing and $\theta *$. Given the law for descriptions, the properties of identity will tell us that

$$\theta(\iota x\ \rho x) \Leftrightarrow (\exists z\colon \rho z \wedge (\forall y\colon \rho y)\ z = y)\ \theta z \vee ((\forall x\colon \rho x)\ (\exists y\colon \rho y)\ \neg\, x = y \wedge \theta *)$$

and the right-hand side is a more formal version of the disjunction used in 8.4.2.

Glen Helman 25 Aug 2005

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 what is said and the role of such a claim as a premise or conclusion. Instead, we need to account for what a definite description refers to.

Conclusions about what a definite description refers to will be relevant to the derivation and the law for the description operator provides a way to draw some. 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 and then using proof by choice for its existential first disjunct; the remaining non-atomic sentences are universals so we cannot expect to go further in a single step. We will call this rule ***Securing a Description*** (SD).

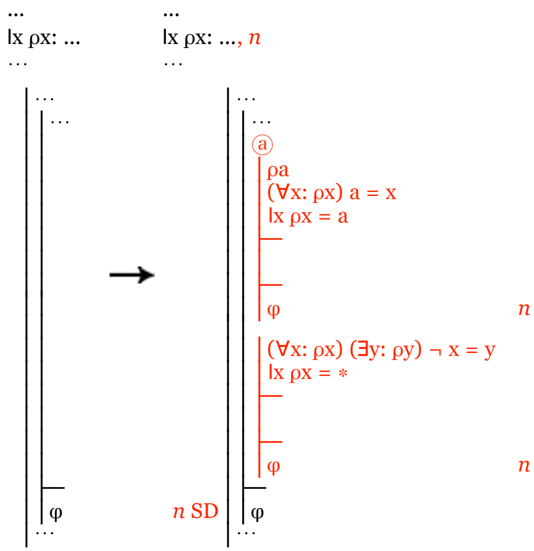


Fig. 8.6.2-1. Developing a derivation at stage *n* by securing a definite description; the parameter a is new to the derivation.

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.

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)	(11)
	ⓐ	
	Wa	(5), (8)
	ⓑ	
	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 SB	(∃y: Wy) ¬ a = y	6
	ⓒ	
	Wc	(8)
	¬ a = c	(9)
	¬ W(Ix Wx)	
8 Adj	Wc ∧ Wa	X, (9)
9 REG	(∃y: ¬ y = c) (Wc ∧ Wy)	X, (10)
10 EG	∃x (∃y: ¬ y = x) (Wx ∧ Wy)	X, (11)
	•	
11 Nc	⊥	7
7 IP	W(Ix Wx)	6
6 PRCh	W(Ix Wx)	3
3 SD	W(Ix Wx)	2
2 CP	Wa → W(Ix Wx))	1
1 UG	∀y (Wy → W(Ix Wx))	

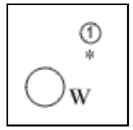
The list of alias sets in the first gap includes ∗ even though that term does not appear in either resources or goals of the gap because, when using the description operator, ∗ is part of our logical apparatus and is thus always among the terms.

Notice that both the premise and the hedge 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 and a derivation showing this provides another example of the use of SD.

ⅰx Wx: 1

	¬ W(ⅰx Wx)	(2)
	ⓐ	
	Wa	(2)
	(∀y: Wx) a = x	
	ⅰx Wx = a	(ⅰx Wx)—a
	•	
2 Nc=	⊥	
	(∀y: Wx) (∃y: Wy) ¬ x = y	∗:3
	ⅰx Wx = ∗	(ⅰx Wx)=∗
	¬ W∗	
	○	¬ W(ⅰx Wx), ⅰx Wx = ∗, ¬ W∗ ⇏ ⊥
	⊥	4
4 IP	W∗	3
	(∃y: Wy) ¬ ∗ = y	
	(unfinished)	
	⊥	3
3 MCR	⊥	1
1 SD	⊥	



The sentence *The winner didn't win* is consistent also on Russell's analysis provided we interpret it as ¬ *the winner won*, for ¬ φ is absurd if and only if φ is a tautology. 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 the property of being a winner and not winning.

Our stipulations about the interpretation of definite descriptions insure that any interpretation of the vocabulary in ρ will divide one of the two gaps that result from SD—that's why there is no precondition for its application—so the rule is utterly sound and its addition will not disturb the soundness of our system. It 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 is divided by an interpretation—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.

However, it is easy to see the completeness of a system of derivations that allows 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:

$$\forall w_1 \ldots \forall w_n\ (\,(\exists z\!: \rho z \wedge (\forall y\!: \rho y)\ z = y)\ ⅰx\ \rho x = z \vee ((\forall x\!: \rho x)\ (\exists y\!: \rho y)\ \neg\, x = y \wedge ⅰx\ \rho x = *)\,)$$

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 ⅰx 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, and this can be insured by the truth of the meaning postulates for the descriptions actually appearing in the argument. That is, if Δ includes the meaning postulate for each description 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.

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 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 parameter, the rule SD can prevent gaps from reaching a dead end. It can be modified to search for finite gaps in the way we have done for other rules using parameters, and named following the same pattern as with those rules as ***Securing a Description Supplemented*** (SD+).

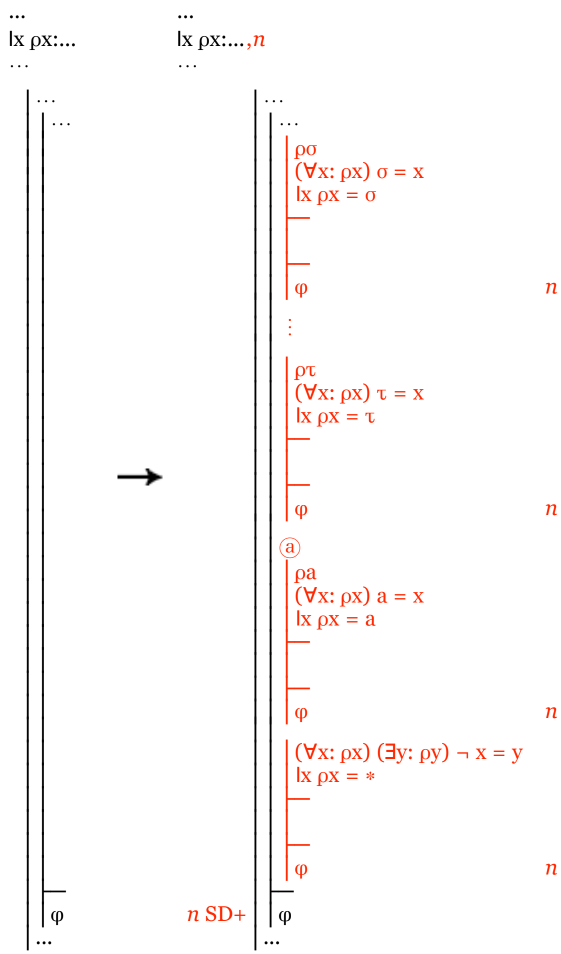


Fig. 8.6.2-2. Developing a derivation at stage *n* by securing a definite description; the parameter a is new to the derivation and the terms σ, ..., τ include at least one from each current alias set for the gap.

Here 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 value might end up as the reference of a definite description. This will happen not only when the description fails to be uniquely satisfied but also when the nil value is the one value satisfying it uniquely. Indeed, the reference of any term τ will uniquely satisfy the predicate λx x = τ, so even if λx (x *is a* C) is not uniquely satisfied λx (x = *the thing that is a* C) will be—though, of course, only by the nil value.

Glen Helman 25 Aug 2005

8.6.s. Summary

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 value. This serves as the reference value of the constant ∗ and as the reference value of the description ℩x ρx when the extension of ρ is empty or has more than one member. Then the law for descriptions asserts that either ℩x ρx is something that is the sole thing ρ is true of or ρ is not true of exactly one thing and ℩x ρx has the nil value.

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). This rule is enough to enable us to establish meaning postulates, which state that definite descriptions are interpreted as we intend. Allow the argument used for completeness of the system of derivations no longer applies, is it 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 when searching for finite counterexamples, it should be used in the alternative form Securing a Description Supplemented (SD+).

Glen Helman 25 Aug 2005

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

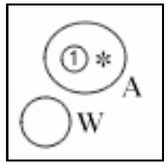
8.6.xa. Exercise answers

1.

	(∃x: Wx ∧ (∀y: ¬ y = x) ¬ Wy) Ax	1
	ⓐ Wa ∧ (∀y: ¬ y = a) ¬ Wy	2
	Aa	(3)
2 Ext	Wa	(3)
2 Ext	(∀y: ¬ y = a) ¬ Wy	
3 REG	(∃x: Ax) Wx	X, (4)
	•	
4 QED	(∃x: Ax) Wx	1
1 PRCh	(∃x: Ax) Wx	

℩x Wx: 3

	A(℩x Wx)	(2)
	(∀x: Ax) ¬ Wx	℩x Wx:2
2 SB	¬ W(℩x Wx)	
	ⓐ Wa	
	(∀x: Wx) a = x	
	℩x Wx = a	(℩xWx)—a, ∗
	•	
3 Nc=	⊥	
	(∀x: Wx) (∃y: Wy) ¬ x = y	∗:4
	℩x Wx = ∗	
	¬ W∗	
	○	A(℩x Wx),¬ W(℩x Wx), ¬ W∗,(℩xWx)=∗ ⇏ ⊥
	⊥	5
5 IP	W∗	4
	(∃y: Wy) ¬ ∗ = y	
	(unfinished)	
	⊥	4
4 MCR	⊥	
3 SD	⊥	1
1 PRCh	(∃x: Ax) Wx	



2.

	(∃x: Ax) Wx	1
	¬ ∃x (∃y: ¬ y = x) (Wx ∧ Wy)	(11)
	ⓐ	
	Aa	(3)
	Wa	(4), (8)
	(∀x: Wx ∧ (∀y: ¬ y = x) ¬ Wy) ¬ Ax	a:3
3 SC	¬ ((Wa ∧ (∀y: ¬ y = a) ¬ Wy))	4
4 MPT	¬ (∀y: ¬ y = a) ¬ Wy	5
	ⓑ	
	¬ b = a	(9)
	Wb	(8)
		(9)
8 Adj	Wa ∧ Wb	X
9 REG	(∃y: ¬ y = a) (Wa ∧ Wy)	X, (10)
10 EG	∃x (∃y: ¬ y = x) (Wx ∧ Wy)	X, (11)
	●	
11 Nc	⊥	7
7 RAA	¬ Wb	6
6 RUG	(∀y: ¬ y = a) ¬ Wy	5
5 CR	⊥	2
2 RNcP	(∃x: Wx ∧ (∀y: ¬ y = x) ¬ Wy) Ax	1
1 PRCh	(∃x: Wx ∧ (∀y: ¬ y = x) ¬ Wy) Ax	

℩x Wx: 2

	(∃x: Ax) Wx	1
	¬ ∃x (∃y: ¬ y = x) (Wx ∧ Wy)	(11)
	ⓐ	
	Aa	(4), (5)
	Wa	(3), (8)
	ⓑ	
	Wb	
	(∀y: Wy) b = y	a:3
	℩x Wx = b	a, b—℩xWx, ∗
3 SB	b = a	a—b—(℩xWx), ∗
	●	
4 QED=	A(℩x Wx)	2
	(∀x: Wx) (∃y: Wy) ¬ x = y	a:5
	℩x Wx = ∗	a, c, (℩xWx)—∗
5 SB	(∃y: Wy) ¬ a = y	6
	ⓒ	
	Wc	(8)
	¬ a = c	(9)
	¬ A(℩x Wx)	
8 Adj	Wc ∧ Wa	X, (9)
9 REG	(∃y: ¬ y = c) (Wc ∧ Wy)	X, (10)
10 EG	∃x (∃y: ¬ y = x) (Wx ∧ Wy)	X, (11)
	●	
11 Nc	⊥	
7 UP	A(℩x Wx)	6
6 PRCh	A(℩x Wx)	2
2 SD	A(℩x Wx)	1
1 PRCh	A(℩x Wx)	