

6.3.3. Derivations for identity

We are now in a position to state the derivation rules for identity that will be part of our basic system. We will have four rules for closing gaps, two pairs of extensions of the rules QED and Nc. One pair is based on the law for co-aliases alone and the other also rests on the laws of congruence for predicates.

The first two rules—*Equated Co-aliases* (EC) and *Distinguished Co-aliases* (DC)—are shown in Figures 6.3.3-1 and 6.3.3-2.

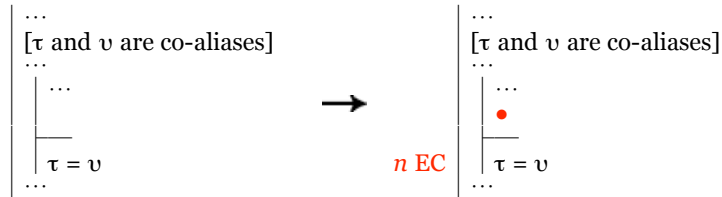


Fig. 6.3.3-1. Closing a gap whose goal is an equation between terms that are co-aliases with respect to the available resources.

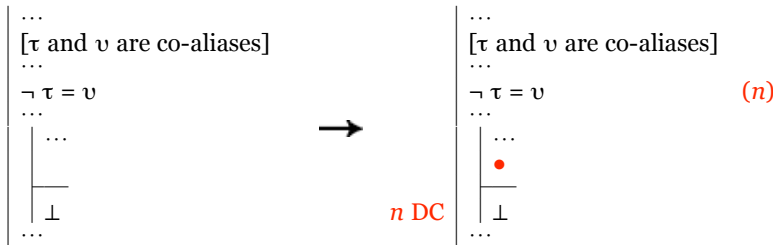


Fig. 6.3.3-2. Closing a gap of a *reductio* argument one of whose resources negates an equation between terms that are co-aliases with respect to the available resources.

The bracketed remark concerning τ and ν stipulates that there be enough equations among the available resources to make the terms τ and ν co-aliases. The law for aliases then tells us that the resources entail the equation $\tau = \nu$. So, if this equation is our goal, we may count the gap closed; and, if its denial is among our resources, we have the inconsistency required to close the gap of a *reductio* argument. An important special case of this rule is one where τ and ν are the same term. In this case, $\tau = \nu$ is $\tau = \tau$ (which is also $\nu = \nu$) and, since a term is a co-alias of itself with respect to any set—even with respect to a set in which it does not appear—any gap with a self-equation as its goal may be closed as may the gap of a *reductio* argument with a negated self-equation among its resources. Notice that the general form of these rules differs from the special case for self-equations only by exchanging terms that are co-aliases.

Some abbreviated terminology will help in stating the next rules for identity. Let us say that two series of terms $\tau_1 \cdots \tau_n$ and $\nu_1 \cdots \nu_n$ are **co-alias series** when they have the same length and their corresponding members are co-aliases—that is, when τ_i and ν_i are co-aliases for i from 1 to n where n is the length of the two series. Then the second pair of rules for identity are shown in Figures 6.3.3-3 and 6.3.3-4. These are **Quod Erat Demonstrandum Given Equations** (QED=) and **Non-contradiction Given Equations** (Nc=).

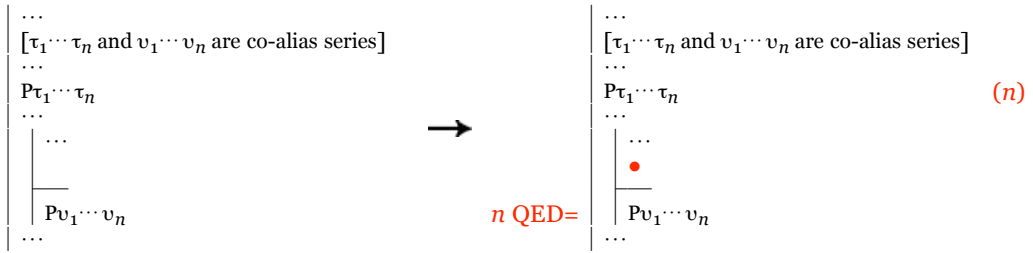


Fig. 6.3.3-3. Closing a gap one of whose resources differs from its goal only by terms that are co-aliases.

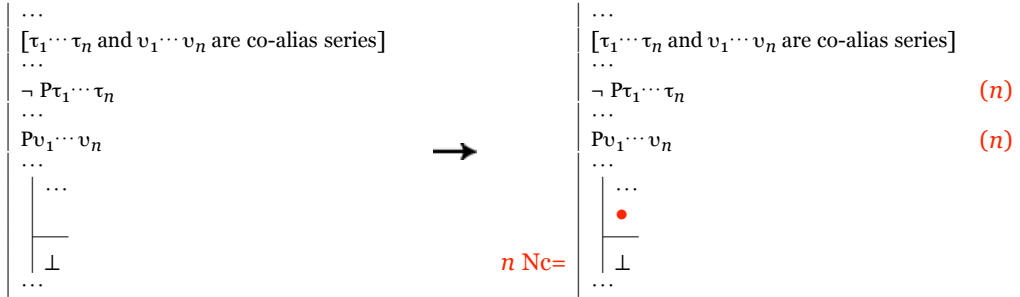


Fig. 6.3.3-4. Closing a gap of a *reductio* argument one of whose resources differs from the negation of another only by terms that are co-aliases.

Here a bracketed remark stipulates that the available resources contain enough equations to make corresponding component terms of $P\tau_1 \dots \tau_n$ and $P\nu_1 \dots \nu_n$ co-aliases and thus to entail identities between these terms. The law of congruence for P then tells us that $P\nu_1 \dots \nu_n$ is entailed by available resources. If it is our goal, we may close the gap, and we may do so also if the gap is in a *reductio* argument and our resources contain its denial $\neg P\nu_1 \dots \nu_n$.

The sentences $P\tau_1 \dots \tau_n$ and $P\nu_1 \dots \nu_n$ that figure in the last two rules have been described as applications of the same predicate whose corresponding component terms are co-aliases. A little thought will show that we can describe such expressions equally well as atomic sentences that differ only by components that are co-aliases. This makes the similarity of this rule to QED and Nc a little more apparent. Instead of saying that we can close a gap when our goal is among our resources and when one resource negates another (as we do in QED and Nc), we say here that we can close a gap if a resource differs from the goal or from the negation of another resource only by co-aliases. This way of describing these rules leads to the question whether we really need to limit them to predications. The answer is that we do not although that is the only case where we really need to use the rule.

Other rules can be extended in the way QED and Nc are extended in QED= and Nc=: if the illustration of the rule displays two occurrences of a sentence, these may be sentences that are different but that differ only by terms that are co-aliases given the available resources. (As was noted above, even our first two rules for identity could be seen as the result of extending in this way rules that say that we can close a gap whose goal is a self-equation and a *reductio* gap whose resources contain the denial of a self-equation.) When is extended in this way, the rule label should be followed by the equals sign, as in the labels for QED= and Nc=. We will call the result an **extension** of the rule **for equations**; and, as with QED= and Nc=, the equals sign added to the name may be read “given equations.”

Below are two derivations that illustrate these ideas. The first uses the rule of Figure 6.3.3-3 to close the gap after stage 2. The second uses an extended version of *modus ponens*; the resource Rc(fa) and the antecedent of $Rcc \rightarrow Gc$ differ only by terms (fa and c) that are co-aliases given the equations $fa = b$ and $b = c$. When an equation is added to the resources, the resulting alias sets are indicated to the right of the equation by listing the members of each with dashes between, separating the members of different alias sets by commas.

$ \begin{array}{l} \text{1 Ext} \\ \text{1 Ext} \\ \\ \\ \text{3 QED=} \\ \text{2 CP} \end{array} \left \begin{array}{l} \hline \text{Fb} \wedge \text{a} = \text{b} \quad 1 \\ \text{Fb} \\ \text{a} = \text{b} \quad (3) \\ \hline \text{a} = \text{d} \quad \text{a-b-d} \\ \bullet \\ \hline \text{Fd} \quad 2 \\ \hline \text{a} = \text{d} \rightarrow \text{Fd} \end{array} \right. $	$ \begin{array}{l} \text{1 Ext} \\ \text{1 Ext} \\ \\ \text{3 MPP=} \\ \text{4 QED} \\ \text{2 CP} \end{array} \left \begin{array}{l} \hline \text{Rc}(\text{fa}) \wedge \text{fa} = \text{b} \quad 1 \\ \text{Rcc} \rightarrow \text{Qc} \quad 3 \\ \hline \text{Rc}(\text{fa}) \\ \text{fa} = \text{b} \quad (3) \\ \hline \text{b} = \text{c} \quad \text{a, fa-b-c} \\ \hline \text{Qc} \quad (4) \\ \bullet \\ \hline \text{Qc} \quad 2 \\ \hline \text{b} = \text{c} \rightarrow \text{Qc} \end{array} \right. $
---	---

If the extended form of *modus ponens* were not used in the second derivation, we would need to set up an indirect proof to reach the goal Qc and exploit $Rcc \rightarrow Qc$ using the rule for exploiting conditionals in *reductio* arguments. Both gaps of the derivation would then close using the identity rules for closing gaps.

The point of listing alias sets to the right of equations is to sum up the co-aliases at each stage when they change. When several equations are added at the same stage (or appear among the initial premises), only the last need be notated in this way. There is no need record the alias sets until the first stage when an equation appears as a resource since up to that point each term is in an alias set by itself. Although it is usually by adding equations that the alias sets will change, this is not the only possible way. When a term is added, either in a new resource or in a new goal, it must be accommodated in the co-aliases sets. Although new terms will be introduced regularly in later chapters, they could be introduced now only if attachment rules or the rule LFR were used to introduce sentences that are not already components of sentences in the derivation. Since that is not a use of such rules that we have been considering, we will not consider examples, but a general guideline for listing alias sets can be stated that will include such cases: at the initial stage of a derivation if it equations as resources and at any stage thereafter at which the alias sets of a gap have changed, list the alias sets at the right near the top of the gap. When several resources are added, the alias sets can be added after the last new resource that figures in the change. If no new resources are added (and the alias sets change only because of vocabulary added in a new goal), the alias sets may be listed at the right of the top of the scope line of the gap.

It is sometimes useful to be able to enter an equation between co-aliases as a further resource. Since this does not change the alias sets, it does bring a gap near an end and it is not automatically progressive. Therefore, we will count it as an attachment rule. We will label this rule by reversing the name of one of the rules for closing gaps: it will be called **Co-alias equation** (CE):

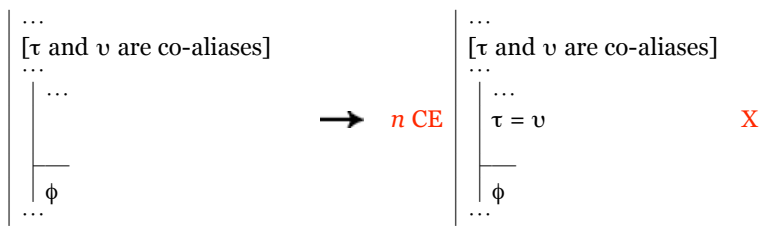


Fig. 6.3.3-5. At stage n , adding an equation between terms that are co-aliases with respect to the available resources.

Equations are never exploited, so the X at the right does not mark the added equation as already exploited; instead it indicates that the equation leads to no change in the alias sets since its component terms are already co-aliases. Since any use of such an equation to close a gap is already covered by other rules, this rule will serve primarily to provide auxiliary resources for detachment rules (and available resources for use in other attachment rules). Here is a simple example.

	a = b		
	b = c		a—b—c, fa—fc, d
	(fa = fc ∧ d = d) → Ga		4
1 CE	fa = fc		X,(3)
2 CE	d = d		X,(3)
3 Adj	fa = fc ∧ d = d		X,(4)
4 MPP	Ga		(5)
	•		
5 QED	Ga		

If we were to use only basic rules, we would need to resort to a *reductio* argument and the rule RC.

The rule CE is really only needed when no co-alias of the terms being equated appears as a term of an equation. The rule would not have been necessary in the example above if the conditional's antecedent had been something like $a = c \wedge b = b$ because this sentence could be added by the extended attachment rule Adj= since it differs from a conjunction of the first two premises only by co-aliases. In general, if our resources contain any equation between co-aliases of the terms that we want to join in the new equation, we have an equation differing from the one we want only by co-aliases and we can use the extended form of whatever rule we might apply to the new equation.

Finally, we will add an attachment rule that allows a resource to be added when it differs from an available resource only by co-aliases. Such resources can be represented as $\theta\tau_1 \cdots \tau_n$ and $\theta v_1 \cdots v_n$ where $\tau_1 \cdots \tau_n$ and $v_1 \cdots v_n$ are co-alias series. That is, it is understood that the differences between the resources are limited to the displayed series of terms so the resources amount to predications to the two series $\tau_1 \cdots \tau_n$ and $v_1 \cdots v_n$ of an abstract θ that takes the form $\lambda x_1 \cdots x_n (\dots x_1 \cdots x_n \dots)$, where $x_1 \cdots x_n$ is a series of distinct variables with the same length as $\tau_1 \cdots \tau_n$ and $v_1 \cdots v_n$. The name of the rule is **Congruence** (Cng).

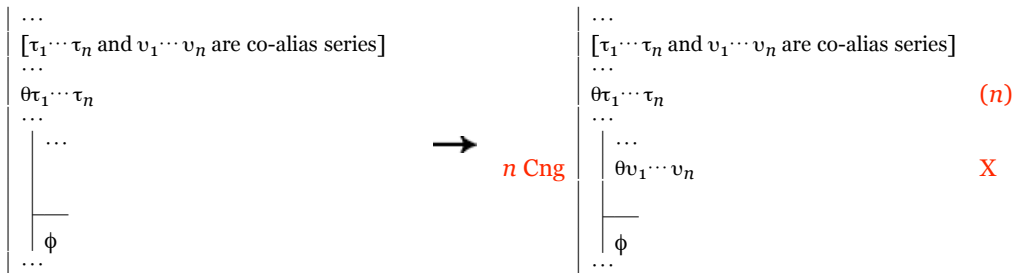


Fig. 6.3.3-6. At stage n , that differs from an available resource only by the occurrence of terms that are co-aliases.

The resource that is added by this rule is marked as exploited because any exploitation of the earlier resource will be sufficient to insure that we have taken account of this new resource. The rule Cng provides the following alternative approach to an earlier derivation.

	Rc(fa) ∧ fa = b		1
	Rcc → Qc		4
1 Ext	Rc(fa)		(3)
1 Ext	fa = b		a, fa—b, c
	b = c		a, fa—b—c
3 Cng	Rcc		X,(4)
4 MPP	Qc		(5)
	•		
5 QED	Qc		2
2 CP	b = c → Qc		

Notice that the ordinary form of MPP is used here rather than the extended form MPP= used earlier, and Cng can always be avoided by using the extended forms of other rules. The point of using Cng is only that it may make it easier to follow an argument.

Glen Helman 23 Oct 2004