6.3. Arguments involving equations

6.3.0. Overview

The basic principles of entailment for identity are among the most familiar of logical principles; but, because equations do not have sentential components, they will play a role in derivations that is quite different from other logical forms we study.

6.3.1. Logical properties of identity
For our purposes, identity amounts to sameness in all respects, a sameness that implies interchangeability as input for any predicate or functor.

6.3.2. A law for aliases
Many of the valid conclusions from a group of equations can be captured by rules telling when terms count as “co-aliases”—i.e., aliases for the same thing.

6.3.3. Derivations for identity
The key rules for identity are rules for closing gaps, but all rules can be extended to reflect the interchangeability of co-aliases.

Glen Helman 22 Oct 2004

6.3.1. Logical properties of identity

The logical properties of identity come from two sources. One is the kind of extension we have stipulated for this relation, the pairs of reference values we say it is true of. The other is the requirement that predicates and functors be extensional, that the compounds they form be transparent to the reference values of their component terms. Properties deriving from this second source are equally properties of the operations of predication and functional application; but since they are not properties of any particular predicate or functor, it is easiest to ascribe them, along with properties of the first sort, to the logical constant =. We will turn first to the properties of identity alone.

What do we know when we know that an equation $\tau = \upsilon$ is true? Well, we know that the terms τ and υ have the same reference value; loosely speaking, we know that they name the same thing. (This is loose speech, first, because the terms may not be names but rather definite descriptions and, second, because the reference value of the terms may be nil, in which case neither names anything.) So we might say that τ and υ are each “aliases” of their common reference value. It will be convenient to have a way of speaking of such terms in relation to each other rather than in relation to their value, so let us say that they are aliases in relation to each other—or, more briefly, that they are ***co-aliases***.

This leads us immediately to a property of identity. For the relation of having the same reference value, of being co-aliases, is ***symmetric***. It does not order the two terms in any way; if we can assert it of them taken in one order, we can equally we assert it of them the other way around. This gives us our first principle for =, the ***law of symmetry for identity***:

$$\tau = \upsilon \Rightarrow \upsilon = \tau.$$

This principle is stated as an entailment, but it implies that the two equations are equivalent since it licenses reversals of equations and we can undo a reversal by reversing again.

Now suppose that we know not only that a term υ is an alias for a term τ but also that τ is an alias for a term σ. All three terms must then have the same reference value, so we could say that υ is an alias for σ. Putting this more formally, we have a second principle, the ***law of transitivity for identity***:

$$\sigma = \tau, \tau = \upsilon \Rightarrow \sigma = \upsilon.$$

Again, there is a more symmetric principle lurking in the background—namely, that any two of these three equations entails the third. But this principle is harder to state compactly, and a fuller investigation of it would show that it also relies on the law of symmetry.

The two laws we have stated tell us that certain equations are true *if* others are, but they do not commit us categorically to the truth of any equations at all. How do we know there are any true equations? Well, what would it take for there to be none. Perhaps this would be so if were no aliases in the ordinary sense and every reference value was the extension of at most one term. We do not want to rule this out, for our laws are supposed to be very general and should not make any assumptions about the richness of our non-logical vocabulary. But even in a case like this, if there is any term at all in our language, we can form an equation with this term taken twice and the equation will be true. And that is one way of stating the ***law of reflexivity for identity***:

$$\Rightarrow \tau = \tau.$$

So there will be true equations if there are any equations at all.

We have found three properties of identity that derive from the kind of extension we have stipulated for =. Collecting them, we have:

Reflexivity. $\Rightarrow \tau = \tau$.
Symmetry. $\tau = \upsilon \Rightarrow \upsilon = \tau$.
Transitivity. $\sigma = \tau, \tau = \upsilon \Rightarrow \sigma = \upsilon$.

Identity is not the only predicate that has these properties. For example, the predicate λx (x *has the same shape as* y) obeys analogous laws; and that example should suggest many others. A predicate for which laws of reflexivity, symmetry, and transitivity hold is said to express an ***equivalence relation*** because such a relation attributes sameness in some respect. For example, if we grant that a line is parallel to itself, we can say that λx (x *is parallel to* y) expresses an equivalence relation; and this relation attributes sameness in direction.

An extreme example of an equivalence relation is the relation that holds between any pair of reference values (including any reference value and itself). Since this relation never fails to hold there is no way for it to violate any of the three laws, and it must be an equivalence relation. The extension of the identity predicate is at the other extreme of equivalence relations. If we represent the two in tabular form as truth-valued functions of reference values, we have something like this.

	0	1	2	3	⋯
0	T	T	T	T	⋯
1	T	T	T	T	⋯
2	T	T	T	T	⋯
3	T	T	T	T	⋯
⋮	⋮	⋮	⋮	⋮	⋱

=	0	1	2	3	⋯
0	T	F	F	F	⋯
1	F	T	F	F	⋯
2	F	F	T	F	⋯
3	F	F	F	T	
⋮	⋮	⋮	⋮		⋱

The first relation has **T** everywhere while the extension of = has **T** only along the diagonal from the upper left to the lower right. Identity holds in the fewest cases possible for an equivalence relation because, if any of the pairs along the diagonal were dropped, the law of reflexivity would not hold. Since identity thus expresses the narrowest equivalence relation, we might think of it as expressing sameness in all respects.

Although the status of identity as the narrowest equivalence relation derives from the extension we have stipulated for =, this does not provide a property that we can express in laws for = alone. Our ability to express the idea of sameness in all respects depends on the predicates and functors we have available to express a variety of "respects." What we can say is that identity implies sameness with regard to each predicate and functor, and we can find further properties of identity by exploiting the consequences of idea. First consider a one-place predicate —say λx (x *is red*). Two things are the same with respect to redness if both are red or neither is. Hence, to say that identity implies sameness with respect to this predicate is say that an equation $\tau = \upsilon$ implies that τ *is red* and υ *is red* have the same truth value. We have no very good way of expressing this sort of relation among three sentences directly, but the symmetry of = means that it is enough to say that $\tau = \upsilon$ and τ *is red* together entail υ *is red*. Generalizing this to any one-place predicate F leads us to assert the law

$$\tau = \upsilon, F\tau \Rightarrow F\upsilon.$$

This is as at least part of what is involved in saying that identity implies sameness in all respects. In fact, if we put θ in place of F and thus allow the predicate to be an abstract, this law says it all. But, for the moment, we will consider only predicates that are not abstracts and say that an equivalence relation that supports a law of this form for a given predicate F is a ***congruence for*** F. An equivalence relation that implies sameness with respect to redness (e.g., the extension of λx (x *has the same color as* y)) is thus a congruence for λx (x *is red*). (The source of the term *congruence* is the geometrical relation of congruence, which implies sameness with respect to size and shape though not with respect to location.)

The form of this law ought to suggest something that is familiar from elementary algebra, the use of an equation to replace one expression by another. Now, in algebra we can equally well use more than one equation to make several replacements simultaneously, and congruence principles can take a similar form. Consider sameness with respect to the relation expressed by a 2-place predicate such as λx (x *is younger than* y). Things that are the same in this respect should be younger than the same things and have the same things younger than them. We can express this idea compactly by the following:

$$\tau_1 = \upsilon_1, \tau_2 = \upsilon_2, \tau_1 \textit{ is younger than } \tau_2 \Rightarrow \upsilon_1 \textit{ is younger than } \upsilon_2$$

And we can claim this holds for 2-place predicates generally by stating the law

$$\tau_1 = \upsilon_1, \tau_2 = \upsilon_2, R\tau_1\tau_2 \Rightarrow R\upsilon_1\upsilon_2.$$

In these statements, we have economized by speaking of both places of the predicate in a single law. Since τ_1 and υ_1 could be the same term and so could τ_2 and υ_2, the law covers cases where a change is made in only one of the two places of R. An equivalence relation that supports a law like this one for identity is said to be a ***congruence for*** the predicate R. The relation of having the same age will be a congruence in this sense for the relation expressed by λx (x *is younger than* y).

Now it should be clear that we might state a law like these two that applies to identity and a predicate P with any number of places:

Congruence for P. $\tau_1 = \upsilon_1, \ldots, \tau_n = \upsilon_n, P\tau_1\ldots\tau_n \Rightarrow P\upsilon_1\ldots\upsilon_n$ (where P has n places).

A large part of what we mean by saying that identity implies sameness in all respects can be captured by saying that it is a congruence for all predicates.

A large part, but not all. We have not yet said anything about functors. Here we can make the story short because the law we want is more familiar. It is this:

Congruence for f. $\tau_1 = \upsilon_1, \ldots, \tau_n = \upsilon_n \Rightarrow f\tau_1\ldots\tau_n = f\upsilon_1\ldots\upsilon_n$ (where f has n places).

This says that an equation between compound terms $f\tau_1\ldots\tau_n$ and $f\upsilon_1\ldots\upsilon_n$ follows from equations between their corresponding components. We can have laws like this for equivalence relations besides identity; and, when we have such a law for an equivalence relation, the relation is said to be a ***congruence for*** the functor f. The relation of having the same absolute value (i.e., of being equal or differing only in sign) is a congruence for a functor expressing the squaring function (or the cosine function). In the case of identity, we can claim congruence for *all* functors.

Have we now captured the properties of identity by saying that it is a congruence for all predicates and all functors? The laws we have stated suffice to capture all true general principles of entailment involving identity, and that was our aim. We might still ask whether a relation could obey these laws without being a relation of sameness in all respects. The question comes to something like this: are the features of a thing that are expressible by predicates and functors sufficient to pin down its identity, to distinguish it from all other things? This is a puzzling question. While any given collection of predicates and functors can certainly fail to express differences among things, it is hard to pin down the claim that there could be such differences that are expressible by no predicates or functors whatsoever, for any attempt to say what such differences might be would begin to undercut the claim that they are inexpressible. In any case, asserting the laws above for all predicates and functors suffices to establish all general principles of entailment concerning identity that we can express using our analysis of logical form.

In saying that identity is a congruence for predicates and functors, we say that predicates and functors are extensional operations and, in particular, that they form referentially transparent compounds. For example, if we were to count the sentence-with-a-gap *For the past two centuries, ____ has been over 35* as a predicate, we could not say that identity is a congruence for all predicates because to assert congruence for this incomplete expression would be to assert the validity of the argument

For the past two centuries, the U. S. president has been over 35
The U. S. president = *George Bush*
___
For the past two centuries, George Bush has been over 35

and, as was noted in 6.1.2, this is naturally understood to have true premises and a false conclusion.

This raises a wider philosophical and logical issue. Could we at least say that this sentence-with-a-blank has an extension that is a function? Such a function would have to yield truth values as output based on something beyond the reference values of the terms to which it was applied, and we might speak of it as an ***intensional property*** (as distinct from as a ***property in intension***, which is merely the way the extensional property expressed by an ordinary predicate varies from world to world). So one part of the question we have just asked is whether there are intensional properties.

The other part is whether there is anything for an intensional property to be a property of. It cannot be a property of an object thought of as a reference value because it depends on distinctions that are ignored in saying what reference value a term has. One way of putting this side of the issue is to ask whether there is any sense of *thing* in which the terms *the U. S. president* and *George Bush* could be said to signify different things. Perhaps we could say that one signifies a public official and the other signifies a person and say that one and the same public official could be identical with different people at different times. The oddity of this talk suggests that nasty problems might lurk here, so we will not open this door any wider. Suffice it to say that logicians and philosophers have adopted a full range of positions on this issue. Some happily accept ***intensional entities*** (such as public officials as distinct from the people who hold those offices) while others reject all talk of intensions, not only of intensional entities and intensional properties but even of the intensions of ordinary extensional predicates.

6.3.2. A law for aliases

If we include identity itself among the predicates and functors for which identity obeys laws of congruence, it will follow that identity is an equivalence relation provided that it is reflexive. For any reflexive relation that is a congruence for itself will be symmetric and transitive also. Moreover, it is enough to say that identity is a reflexive relation and a congruence for all one-place predicates, simple and complex (i.e., including abstracts), to show that identity is a congruence for all predicates and functors whatsoever. This provides one compact way of capturing the laws for identity; but, when implementing these laws in derivation rules, it will be most convenient to group them together in a different way. We will have two principles, one of which will be the law asserting that identity is a congruence for all simple predicates. The other will be a law that groups reflexivity, symmetry, and transitivity together with the law of congruence for functors.

We arrived at the laws of symmetry and transitivity by understanding an equation $\tau = \upsilon$ to say that the two terms τ and υ are co-aliases, and we might have arrived at the law of reflexivity in the same way since in the usage of *alias* we have adopted here each term is a co-alias of itself. We can extend these same ideas more generally by speaking of terms τ and υ as being ***co-aliases given*** a set Δ of equations or as being ***made co-aliases by*** Δ. Our intention is that this relation capture the conditions under which equations are entailed by other equations.

Clearly a set Δ of equations will imply that an equation $\tau = \upsilon$ is true if either an equation between τ and υ (in either order) appears in Δ or one term can be reached from the other via a series of terms, each of which is linked to the next (in either order) by an equation in Δ. For example, if the equations shown in Figure 6.3.2-1 are in a set Δ, the terms a and e are co-aliases given Δ, as are any other pair of terms appearing in the list. Moreover, we may count each term as an alias for itself given any set of equations.

```
a = b
    |
    b = c
        |
    d = c
    |
    d = e
```

Fig. 6.3.2-1. A chain of equations making terms a and e co-aliases.

Although we have not yet stipulated all the conditions under which we will count terms as co-aliases, we have said enough to summarize the laws of reflexivity, symmetry, and transitivity—and more besides—by stating a

> ***Law for aliases***: $\Gamma \Rightarrow \tau = \upsilon$ if τ and υ are co-aliases given the set of equations in Γ.

Like the law for a premise as a conclusion and a number of other principles we have used, the law for aliases gives sufficient but not necessary conditions for an entailment to hold, so it is stated with *if* rather than *if and only if*. To see why an equation can be a valid conclusion without its component terms being made co-aliases by the premises, note that, while an equation will be entailed by a set of equations only if it equates terms made co-aliases by that set, an equation can be entailed by a set of sentences without being entailed by the equations in the set. (For example, t = u is entailed by the premises A → t = u and A, and that set contains no equations at all, only a conditional and an unanalyzed sentence.)

Linking a pair of terms by a chain of equations is not the only way a set might imply that they have the same extension. Recall the law of congruence for an *n*-place functor f

$$\tau_1 = \upsilon_1, \ldots, \tau_n = \upsilon_n \Rightarrow f\tau_1 \ldots \tau_n = f\upsilon_1 \ldots \upsilon_n$$

This tells us that the terms $f\tau_1 \ldots \tau_n$ and $f\upsilon_1 \ldots \upsilon_n$ must have the same extension whenever their corresponding components (i.e., τ_1 and υ_1, τ_2 and υ_2, and so on) do. To incorporate this principle into the law for aliases, we will want to say that two applications of a given functor are made co-aliases whenever their corresponding components are made co-aliases, and we will want to allow this sort of connection between terms to figure as a link in a chain by which further terms are made co-aliases.

Putting all this together, we can give a fuller definition of the idea of co-aliases as follows:

The ***co-aliases given*** a set Δ of equations include pairs of terms of all of the following kinds:

(i) a term paired with itself;
(ii) a pair of terms equated (in either order) by a member of Δ;
(iii) a pair of terms connected by a chain of terms linked as co-aliases given Δ;
(iv) a pair of applications of the same functor whose corresponding components are co-aliases given Δ.

Notice that the third and fourth classes are described in terms of the relation we are defining. A definition like this can be thought as a series of instructions for building the extension of the relation it defines. We first put in all the pairs covered by instructions (i) and (ii). Then we gradually add more pairs as we are directed to by instructions (iii) and (iv), replacing the phrase "co-aliases given Δ" by "pairs already in the extension." A pair of terms then count as co-aliases given Δ if and only if they are added at some stage in this process. And, since this process of building an extension for a two-place predicate can be described without using the term *co-alias*, we really have explained the meaning of that term.

In the simple examples we will usually consider, it will be easy to see which terms are co-aliases given a set of equations. But it may help in understanding the idea to think of the sort of "calculation" we might perform to apply the definition in a more complex example. When checking to see whether a pair of terms τ and υ are co-aliases given a set Δ of equations, let us collect all terms appearing as components in τ, υ, and Δ. Figure 6.3-2 shows these terms for a case where the set Δ consists of the equations a = b, fb = c, fb = fc, d = gca, and g(fa)b = e and we are checking to see whether the terms a and fd are co-aliases.

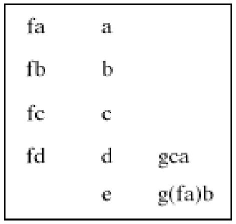
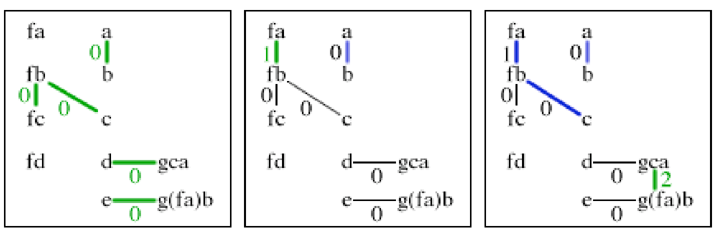


Fig. 6.3.2-2. A work space for finding co-aliases.

Notice that we include fa because it is a component g(fa)b; however, there is no need to include fe or other more complex terms that could be formed from this vocabulary. (The arrangement of the terms is not significant; the one used here is designed simply to make later steps easier to depict.)

Now let us accumulate links between co-aliases. We will represent them as lines between terms. At the initial stage (which we will label 0), we put in links corresponding to equations in the set Δ. We can follow instruction (iii) after this and after each succeeding stage by considering terms to be co-aliases when they are linked either directly or by a chain, so there is no need to draw additional lines. At each of the stages from 1 on, we will consider all functors appearing among the terms and add any links we are directed to by instruction (iv); this will usually require new lines. We may need to do this several times over, but if we add no new links at any stage we can stop because there will be nothing to add thereafter. And, with a finite number of terms, this must happen at some point because there are only a finite number of links we might add.

Figure 6.3.2-3 shows such a process for the example of Figure 6.3.2-2, using labels on links to record the order in which they are entered. The new links at each stage are emphasized along with any older links that lead to the new entry. At stage 1, we check the applications of the functors f and g to see whether we can add any links by instruction (iv). Since a and b were already linked at stage 0, we add a link between fa and fb. We add no other links between the applications of f because d is not linked to a, b, or c. One pair of corresponding terms from gca and g(fa)b (viz., a and b) were connected at stage 0 but the other pair were not, so the link between the two applications of g is entered only at stage 2 after c and fa have also been connected (by the link between fa and fb we enter at stage 1). Even at stage 2 the group including d is not linked to either the groups in which a, b, and c appear, so there are no further links between applications of f and the process is complete. The terms a and fd we were checking do not prove to be co-aliases at the end, but many other pairs of terms were shown to be co-aliases.



Stage 0 Stage 1 Stage 2

Fig. 6.3.2-3. Terms classified as co-aliases in a series of stages.

The links connect the terms in groups shown in Figure 6.3.2-4. The members of any group are co-aliases of one another but not of any other terms.

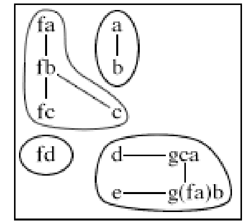


Fig. 6.3.2-4. Linked terms grouped in alias sets.

Some terms, like fd in the diagram, may be groups unto themselves; but, because they are co-aliases of themselves, we can still say that any pair made from such a group is a pair of co-aliases. If the terms had been written down more randomly, the links between them might have crossed and the groups of connected terms would no longer stand out; but they would still be there, and any diagram can be disentangled so that they appear. (This is a distinguishing feature of equivalence relations; any such relation divides a range of values into non-overlapping ***equivalence classes***.) We will refer to each such group of connected terms as an ***alias set***.

Now we are ready to justify our law of aliases, which claims that $\Gamma \Rightarrow \tau = \upsilon$ whenever τ and υ are co-aliases given the equations in Γ. We can do this by showing how this law summarizes earlier ones. Each of the instructions (i)-(iv) for building connections between terms implements one or more of laws of entailment:

	instruction	*law(s)*
i.	enter all terms appearing as components in τ, υ, and the set Δ of equations appearing in Γ	law of reflexivity (since entering the term establishes a link with itself)
ii.	link each pair of terms equated (in either order) by a member of Δ	law for a premise as a conclusion and the law of symmetry (since a link amounts to an equation in both directions)
iii.	count as linked any pair of terms connected by a chain of links	law of transitivity
iv.	link any pair of applications of the same functor whose corresponding components are linked	law of congruence for functors

We combine laws in (ii) and also through carrying out the instructions in a series of stages. This combination of laws can be justified by the law for lemmas because we can think of the process of adding links as a process of adding further equations as lemmas.

Glen Helman 22 Oct 2004

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.

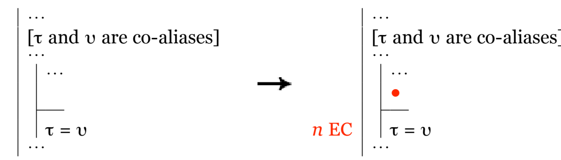


Fig. 6.3.3-1. Closing a gap whose goal is an equation between terms that are co-aliases with respect to the available resources.

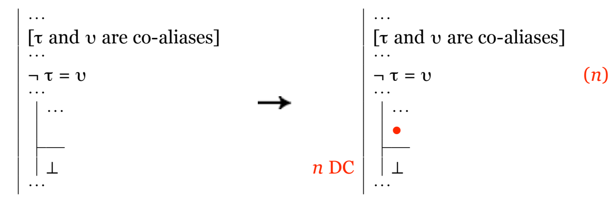


Fig. 6.3.3-2. Closing a gap of a *reductio* argument one of whose resources negates an equation between terms that are co-aliases with respect to the available resources.

The bracketed remark concerning τ and υ stipulates that there be enough equations among the available resources to make the terms τ and υ co-aliases. The law for aliases then tells us that the resources entail the equation τ = υ. 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 υ are the same term. In this case, τ = υ is τ = τ (which is also υ = υ) 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 $\upsilon_1 \cdots \upsilon_n$ are ***co-alias series*** when they have the same length and their corresponding members are co-aliases—that is, when τ_i and υ_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=).

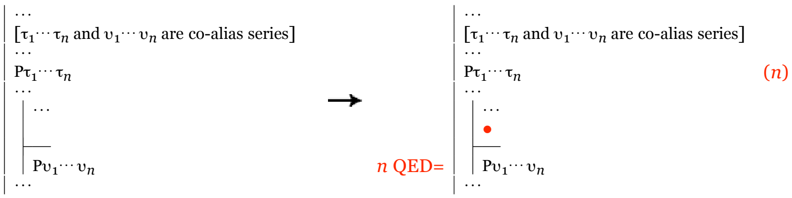


Fig. 6.3.3-3. Closing a gap one of whose resources differs from its goal only by terms that are co-aliases.

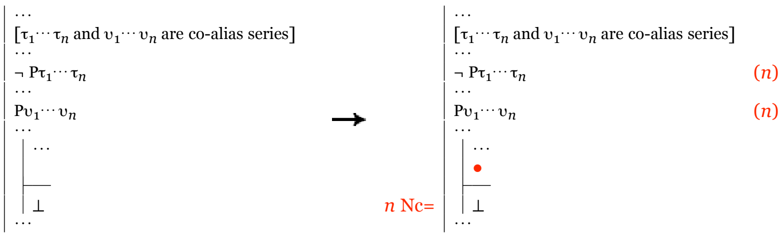


Fig. 6.3.3-4. Closing a gap of a *reductio* argument one of whose resources differs from the negation of another only by terms that are co-aliases.

Here a bracketed remark stipulates that the available resources contain enough equations to make corresponding component terms of $P\tau_1 \cdots \tau_n$ and $P\upsilon_1 \cdots \upsilon_n$ co-aliases and thus to entail identities between these terms. The law of congruence for P then tells us that $P\upsilon_1 \cdots \upsilon_n$ 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 $\neg P\upsilon_1 \cdots \upsilon_n$.

The sentences $P\tau_1 \cdots \tau_n$ and $P\upsilon_1 \cdots \upsilon_n$ 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 → 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.

	Fb ∧ a = b	1
1 Ext	Fb	(3)
1 Ext	a = b	a—b, d
	a = d	a—b—d
	•	
3 QED=	Fd	2
2 CP	a = d → Fd	

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

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 → 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 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-alias 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):

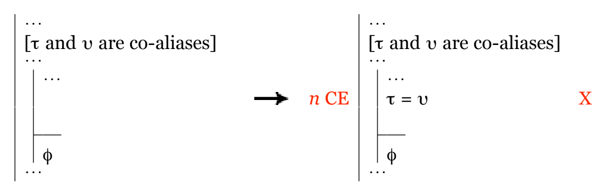


Fig. 6.3.3-5. At stage *n*, adding an equation between terms that are co-aliases with respect to the available resources.

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 ∧ 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 $\theta\upsilon_1\cdots\upsilon_n$ where $\tau_1\cdots\tau_n$ and $\upsilon_1\cdots\upsilon_n$ 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 $\upsilon_1\cdots\upsilon_n$ of an abstract θ that takes the form $\lambda x_1\cdots x_n$ $(\ldots x_1\cdots x_n\ldots)$, where $x_1\cdots x_n$ is a series of distinct variables with the same length as $\tau_1\cdots\tau_n$ and $\upsilon_1\cdots\upsilon_n$. The name of the rule is ***Congruence*** (Cng).

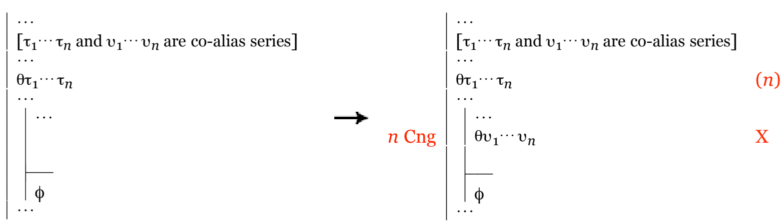


Fig. 6.3.3-6. At stage *n*, that differs from an available resource only by the occurrence of terms that are co-aliases.

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.

6.3.s. Summary

The logical properties of identity have two sources, the extension stipulated for = and the requirement that all predicates and functors be extensional. We will approach these properties by speaking of the terms equated by a true equation as co-aliases. Some thought about this idea shows us that identity obeys laws of reflexivity, symmetry, and transitivity. There are other relations obey laws of the same form; any relation that obeys laws of all three sorts is an equivalence relation and can be thought to ascribe sameness in certain respects. Identity is distinguished as holding in the fewest cases of any equivalence relation; it implies sameness in respect to all predicates and functors. That is, identity is a ***congruence for*** each predicate and functor. All of the laws for identity can be summarized by saying that identity is reflexive and is a congruence for all one-place predicates, abstracts included. To say that identity is a congruence for a predicate or functor is to say that it is an extensional operation. A predicate that did not satisfy this requirement would be an intensional property (as distinct from a property in intension, which is the meaning of an ordinary extensional predicate) and the things of which it was true or false would be intensional entities. Whether these ideas are needed to account for aspects of deductive reasoning (or are even coherent) has been a matter of controversy, but we will consider only extensional operations.

A different way of organizing the laws for identity is useful in stating derivation rules. We say that terms are co-aliases given a set Δ of equations if an equation between the terms follows from Δ. A set Δ of equations serves to divide a collection of terms into alias sets, groups of terms whose members are mutual co-aliases; these are examples of the equivalence classes associated with any equivalence relation. The alias sets determined by a given set of equations can be found by a process of making links between terms, following rules that implement the laws for identity. As a result, identity obeys a law for aliases that says that an equation $\tau = \upsilon$ is entailed by a set of premises if the terms τ and υ are co-aliases given the equations among those premises.

The law for aliases and the law of congruence for predicates provide us with the basic derivation rules for =, each of which is a rule for closing gaps. The rules employ the idea of terms being co-aliases given the equations among the resources of a gap. One, Equated Co-aliases (EC), says a gap may be closed if its goal is an equation between co-aliases, and another, Distinguished Co-aliases, says a *reductio* gap may be closed if its resources include a denial of such an equation. A second pair concern predications of the same predicate to series of terms whose corresponding members are co-aliases. One of these, QED Given Equations (QED=) says that a gap may be closed if its goal is a predication that differs from another predication among the resources only by co-aliases and another, Nc Given Equations (Nc=) says that a *reductio* gap may be closed if one of its resources differs from what another denies only by co-aliases. The statements of these rules use the idea of co-alias series of terms, two series of the same length whose corresponding terms are co-aliases. The idea behind this second pair of rules can be carried further and we may extend any rule by counting as identical, for the purposes of applying the rule, any sentences that differ only by terms that are co-aliases. There are two attachment rules for identity that may be convenient. One, Co-alias Equation (CE), allows us to add to the resources any equation between co-aliases and the other, Congruence (Cng), allows us to add a predication that differs only by co-aliases from one already among the available resources.

Glen Helman 23 Oct 2004

6.3.x. Exercise questions

Use the system of derivations to establish each of the following:

1. Fa → Ga, Fa, a = b ⇒ Gb

2. Fa → Ga, Fb, a = b ⇒ Ga

3. Fa ∧ a = gb ⇒ ¬ F(gc) → ¬ b = c

4. Fa → G(fa), G(fb) → Hb, a = b ⇒ Fb → Ha

5. fa = b, fc = d ⇒ (a = c ∨ b = d) → fa = d

6. *The vice president is Dick Cheney*
George Bush is the president
The vice president is not from Texas
____
If George Bush is from Texas, then Dick Cheney is not the president

Glen Helman 01 Aug 2004

6.3.xa. Exercise answers

Some of the derivations below are given twice, once using only the basic identity rules EC, DC, QED=, and Nc= and a second time using MPP= and similar extensions for equations of other rules (see 6.3.3); either approach is entirely acceptable.

1.

	Fa → Ga	1
	Fa	(1)
	a = b	a—b
1 MPP	Ga	(2)
	•	
2 QED=	Gb	

2.

	Fa → Ga	2
	Fb	(3)
	a = b	a—b
	│ ¬ Ga	(2)
2 MTT	│ ¬ Fa	(3)
	│ •	
3 Nc=	│ ⊥	1
1 IP	Ga	

	Fa → Ga	1
	Fb	(1)
	a = b	a—b
1 MPP=	Ga	(2)
	•	
2 QED	Ga	

3.

	Fa ∧ a = gb	1
1 Ext	Fa	(4)
1 Ext	a = gb	a—gb, b, c, gc
	│ ¬ F(gc)	(4)
	│ │ b = c	a—gb—gc, b—c
	│ │ •	
4 Nc=	│ │ ⊥	3
3 RAA	│ ¬ b = c	2
2 CP	¬ F(gc) → ¬ b = c	

4.

	Fa → G(fa)	3
	G(fb) → Hb	5
	a = b	a—b, fa—fb
	Fb	(4)
	¬ Ha	(7)
	•	
4 QED=	Fa	3
	G(fa)	6
	•	
6 QED=	G(fb)	5
	Hb	(7)
	•	
7 Nc=	⊥	5
5 RC	⊥	3
3 RC	⊥	2
2 IP	Ha	1
1 CP	Fb → Ha	

	Fa → G(fa)	2
	G(fb) → Hb	3
	a = b	a—b, fa—fb
	Fb	(2)
2 MPP=	G(fa)	(3)
3 MPP=	Hb	(4)
	•	
4 QED=	Ha	1
1 CP	Fb → Ha	

5.

	fa = b	
	fc = d	a, b—fa, c, d—fc
	a = c ∨ b = d	2
	a = c	a—c, b—fa—fc—d
	•	
3 EC	fa = d	2
	b = d	a, fa—b—d—fc, c
	•	
4 EC	fa = d	2
2 CP	fa = d	1
1 CP	(a = c ∨ b = d) → fa = d	

6.

	v = c	
	b = p	b—p, c—v
	¬ Fvt	(3)
	Fbt	(3)
	c = p	b—p—c—v
	•	
3 Nc=	⊥	2
2 RAA	¬ c = p	1
1 CP	Fbt → ¬ c = p	

[F: λxy (x *is from* y); v: *the vice president*; b: *George Bush*; c: *Dick Cheney*; p: *the president*; t: *Texas*]

Glen Helman 23 Oct 2004