6.4.3. Structures as counterexamples

Since structures provide the information that is now needed to determine truth values for sentences, we will present counterexamples to derivations that fail by describing structures. An example of a failed derivation is shown below.

```
          │P(fa)b → Qa(fd)          3
          │Qbd → Fb                 5
          │b = d                    a, b—d, fa, fd
          ├──
          │ │P(fd)d ∧ a = d         2
          │ ├──
2 Ext     │ │P(fd)d                 (3)
2 Ext     │ │a = d                  a—b—d, fa—fd
3 MPP=    │ │Qa(fd)
          │ │ │¬ Fd                 (5)
          │ │ ├──
5 MTT=    │ │ │¬ Qbd
          │ │ │○                    b=d,P(fd)d,a=d,Qa(fd),¬ Fd,¬ Qbd ⇏ ⊥
          │ │ ├──
          │ │ │⊥                    4
          │ ├──
4 IP      │ │Fd                     1
          ├──
1 CP      │(P(fb)d ∧ a = d) → Fd
```

Stage 3 of the development uses the extended version of *modus ponens*. At this point, we have two alias sets, one consisting of a, b, and d and the other consisting of fa and fd. We do not have the antecedent of the conditional P(fa)b → Qa(fb) among our resources but rather a sentence, P(fd)d, that, although differing from it in two places, differs only by terms that are co-aliases for fa and b. Stage 5 uses a similarly extended *modus tollens*. The remaining open gap cannot be closed because Qa(fd) and Qbd, the two resources that might be part of a contradiction, differ in their second place by terms (fd and d) that have not been made co-aliases.

The active resources of the dead-end gap form the consistent set:

b = d, P(fd)d, a = d, Qa(fd), ¬ Fd, ¬ Qbd

To describe a structure making the members of this set true, we must choose a range of reference and assign an extension to each of the items of non-logical vocabulary. The choice of the referential range and the assignment of extensions to both individual terms and functors is determined by the alias sets. We choose one reference value for each alias set and assign extensions so that the terms in the set have that as their reference value.

For this consistent set, we will have two alias sets, one containing a, b, and d and the other containing fa and fd, so we take the range to consist of two values, one corresponding to each alias set. We do this by numbering the alias sets and taking these numbers to be the IDs of the values in the range.

Next we must assign values to non-logical vocabulary appearing in the terms in such a way that each term has the reference value corresponding to the number of its alias set. In the case of an unanalyzed term we simply assign the value of its alias set. In the case of a compound term, we place the following constraint on the interpretation of its main functor (the one used last in forming it): the output must be the value associated with the alias set of the compound when the input consists of the reference values associated with the alias sets of the component terms. In the example we are looking at, the two compound terms place the same constraint since they are co-aliases and have components which are co-aliases. The table below shows the association of ID numbers with alias sets and the constraints on the structure that follow from this association:

term	*ID*	*constraint*
a	1	a: 1
b		b: 1
d		d: 1
fa	2	f1: 2
fd		f1: 2

To indicate constraints, we use a variant of the notation used to indicate the extensions of functors in the diagrammatic presentation of structures. Here "f1: 2" says that interpretation of f must yield output with ID 2 for input with ID 1.

We also have three non-logical predicates to consider, the 2-place predicates P and Q and the 1-place predicate F. Each sentence in the consistent set that affirms or denies one of these of a series of terms provides a constraint on the interpretation of that predicate—as is shown in the following table.

resource	*constraint*
P(fd)d	P21: **T**
Qa(fd)	Q12: **T**
¬ Qbd	Q11: **F**
¬ Fd	F1: **F**

The sentence P(fd)d tells us that P is true of values 2 and 1 (in that order) since these are the values of fd and d, respectively; but no other sentence says anything about the extension of P. There are sentences that require that the predicate Q be true of the pair 1 and 2 and false of the pair 1 and 1, but nothing is said about other cases. The last sentence requires that F be false of 1 but requires nothing beyond this.

The tables below incorporate this information about extensions. The values in grey are not required to make the members of the consistent set true and may be assigned arbitrarily. In the case of predicates, the value **F** has been assigned in such cases to make the extension as small as possible.

R: 1, 2

a	b	d
1	1	1

τ	fτ
1	2
2	1

τ	Fτ
1	F
2	F

P	1	2
1	F	F
2	T	F

Q	1	2
1	F	T
2	F	F

The upshot of these tables is depicted in Figure 6.3.4-1.

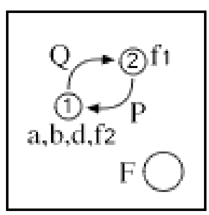

Fig. 6.4.3-1. A structure dividing the open gap of the derivation above.

Since the predicates P and Q are each true of only one pair, they are used to label arrows directly. The emptiness of F's extension is shown by using F to label a circle that encloses nothing. This structure is small enough that the extension of the functor f is also represented in the diagram.

Much of the work here comes in assigning interpretations to individual terms and functors on the basis of a collection of alias sets. Let us look at another example of that. The example we worked out in 6.3.2 would arise if we were to check the entailment

$$a = b,\ fb = c,\ fb = fc,\ d = gca,\ g(fa)b = e \Rightarrow a = fd$$

The derivation for this is not very interesting. A single use of IP would leave us with a dead-end open gap which fails to close because

$$a = b,\ fb = c,\ fb = fc,\ d = gca,\ g(fa)b = e,\ \neg\ a = fd \not\Rightarrow \bot$$

The alias sets we found in 6.3.2 are shown below along with the corresponding constraints on the interpretation of individual terms and functors:

term	*ID*	*constraint*
a	1	a: 1
b		b: 1
c	2	c: 2
fa		f1: 2
fb		f1: 2
fc		f2: 2
fd	3	f4: 3
d	4	d: 4
e		e: 4
gca		g21: 4
g(fa)b		g21: 4

As in the example above, an unanalyzed term is simply assigned the number of its alias set. For a compound term, we require that the number of the alias set be the output value corresponding to input(s) that are the numbers of the alias sets of its immediate components. For example, the term fa appears in set 2, so we want the table for f to lead us to calculate 2 as the reference value of fa. The input for the calculation will be the reference value of the term a; but a appears in set 1, so we want the table for f to yield output 2 for input 1. We derive exactly the same information from the appearance of the term fb; the output is the same because it appears in the same alias set as fa, and the input is the same because the term b appears in the same alias set as the term a. On the other hand, the appearance of fc in alias set 2, tells us that the table for f should assign output 2 also for input 2 since 2 is the alias set of the term c. We respond to the remaining terms in a similar way, the only difference being the need to note pairs of input values in the case of the 2-place functor g.

When we put constraints in tables assigning extensions to the individual terms a, b, c, d, and e the functors f and g, we get the following:

R: 1, 2, 3, 4

a	b	c	d	e
1	1	2	4	4

τ	fτ
1	2
2	2
3	
4	3

g	1	2	3	4
1				
2	4			
3				
4				

Many entries are left unfilled because they did not correspond to any terms in our alias sets. But, by the same token, we will never use these entries to calculate the values of terms appearing in the open gap, so they can be filled in arbitrarily. The value 1 is used in the tables below but any other would do; it is the other values that are significant.

R: 1, 2, 3, 4

a	b	c	d	e
1	1	2	4	4

τ	fτ
1	2
2	2
3	1
4	3

g	1	2	3	4
1	1	1	1	1
2	4	1	1	1
3	1	1	1	1
4	1	1	1	1

Recall that, in a couple of cases, we have had a single input-output pair dictated by two different terms. This raises the question whether the procedure we are using could ever lead to impose incompatible requirements? That is, could we end up trying to associate two different output values of a functor with the same series of input values and thus to fill in one entry in two different ways? For this to happen, there would have to be terms $f\tau_1...\tau_n$ and $f\upsilon_1...\upsilon_n$ with a common functor f that fell into different alias sets (if we were to have two output values), and the corresponding components of these compounds (τ_i and υ_i for i from 1 to n) would have to fall in the same alias sets (if we were to have the same input values in the two cases). But the way we have set up alias sets insures that this cannot happen. Instruction (iv) for drawing links would have told us to put the two compounds in the same alias set once their corresponding components were connected. And, indeed, in the two cases where we have duplicate requirements, the compounds appear in the same alias set precisely because we followed this instruction when forming the alias sets of this example. (Although terms whose corresponding components are co-aliases are bound to appear in the same alias set, they might do so for other reasons, too; for example, we might have both a = b and fa = fb as resources of a gap we are trying to divide.)

We have now done enough to settle the truth values of all equations that appear affirmed or negated among the premises we are trying to make true. Do these values come out as we would like? That is, do the affirmed equations come out true and the negated ones false? Well, since the extensions given to all terms, simple or compound, will correspond to their alias sets, we know that any equation τ = υ that is affirmed among the premises will be true. For such an equation will have led us to put the terms τ and υ into the same alias set, and each term will be assigned the value corresponding to this set as its extension. And, since they have the same extension, the equation between them will be true. How about the denial of an equation, a resource of the form ¬ τ = υ? Since the gap cannot be closed, we know that τ and υ are members of different alias sets. And since the extensions given to these terms correspond to their alias sets, they will have different reference values and the equation $\tau = \upsilon$ will be false, making the resource $\neg\, \tau = \upsilon$ true—as is the case with ¬ a = fd in the example above.

We have been focusing on functors and equations since that is all that matters for the example, but similar considerations apply to non-logical predicates and predications of them. In the case of such predicates, it is our rules for closing gaps insure that we can assign interpretations consistently. If the gap cannot be closed we know that it does not contain both $P\tau_1 \ldots \tau_n$ and any sentence $\neg\, P\upsilon_1 \ldots \upsilon_n$ where the corresponding terms are co-aliases. And this means it never contains both an affirmation and a denial of P of any series of terms whose corresponding members are in the same alias sets. This means that we will never be led to require the extension of P to yield two different outputs for the same input. And the requirements we place on the extensions of non-logical predicates are designed to insure directly the truth of sentences affirming or denying the predication of such a predicate, so it is enough to know that our requirements are consistent to be sure that they will have the desired result.

The procedure we have been following enables us to find a structure dividing any dead-end open gap, and the non-creativeness of our rules tells us that the same structure will divide the initial premises and conclusion of the derivation. Now the existence of a structure dividing premises and conclusion is the test of formal validity of an argument. That is, if there is a structure that divides an argument's premises from its conclusion, then there is an intensional interpretation of it producing an actual English argument and a possible world that will divide the premises and conclusion of that argument. In 2.3.1, we saw why this was so in the case of arguments involving only conjunction, and the same ideas apply to all of truth-functional logic; but more needs to be said in the case of the more complex interpretations we are now considering.

We cannot, as in 2.3.1, simply choose the actual world as the possible world that divides premises from conclusion because a structure, such as the one in Figure 6.4.3-1, may have only a limited number of reference values, while the actual world has many things in it (infinitely many if numbers are counted). The easiest approach in the present setting (but one that will no longer work in the next chapter) is to note that our calculations of extensions for the terms we are interested in remain the same in the presence of further reference values. When we chose a referential range, we could have added reference values that did not correspond to alias sets. Such values would not have played a role in the constraints on the interpretation of non-logical vocabulary or in the calculations of the values of components of the premises and conclusion of the argument we are interested in. So they would have neither contributed to nor interfered with the task of dividing the premises from the conclusion. The possibility of adding such further reference values means that we can regard a structure like that of Figure 6.4.3-1 as a depiction of the way things stand for certain reference values *among others*. Given this understanding of a structure, it is not too hard to concoct intensional interpretations of the non-logical vocabulary that have the right extensions in the actual world. We might, for example, choose language describing an illustration of the structure. To capture the structure of 6.4.3.1, the interpretation of the term a could be *the point labeled a* and the interpretation of P could be λxy (*a P-arrow runs from* x *to* y). If we use this sort of interpretation, drawing the structure is a way of making the actual world divide the argument's premises from its conclusion.

Glen Helman 24 Oct 2004