Thibaut Giraud, 2015, “On Modal Meinongianism”, *Synthese*, DOI: 10.1007/s11229-015-0933-8.

**On Modal Meinongianism**

**Abstract:**

Modal meinongianism is a form of meinongianism whose main supporters are Graham Priest and Francesco Berto. The main idea of modal meinongianism is to restrict the logical deviance of meinongian non-existent objects to *impossible worlds* and thus prevent it from “contaminating” the actual world: the round square is round and not round, but not in the actual world, only in an impossible world. In the actual world, supposedly, no contradiction is true.

I will show that Priest’s semantics, as originally formulated in *Towards Non-Being*, tell us something different. According to certain models (especially models that are interesting from a meinongian point of view), there are true contradictions in the actual world. Berto and Priest have noticed this unexpected consequence and have suggested a solution[[1]](#footnote-2), but I will show that their solution is highly questionable.

In the last section of this paper, I will introduce a new and simpler version of modal meinongianism that avoids the problem.

**1. Presentation of modal meinongianism**

Meinong’s theory of object rests on a *characterization principle* that can be put in these terms: if an object is characterized in a certain way, then it really has the properties through which it is characterized. This unrestricted principle leads to obvious inconsistencies, so it must be restricted in some way. The two main strategies are (i) a distinction between two sorts of properties (see an example, Parsons (1980)) and (ii) a distinction between two modes of predication (see an example, Zalta (1983)).

Priest (2005) defends another sort of meinongianism whose central idea is the following:

“the [characterization principle] can be accepted *in full generality*: we just do not assume that an object characterized in a certain way has its characterizing properties in the actual world, only in the worlds which realize the way the agent represents things to be in the case at hand.” (p.85)

Berto (2008, 2011 and 2013) developed a simplified version of Priest’s theory and to describe it coined the name *modal meinongianism*. The general idea is to “delocalize” meinongian impossible objects to impossible worlds. In impossible worlds, objects can behave in the most illogical way; but, since they do not exist in the actual world, we do not have to endorse a paraconsistent or dialetheist view: in the actual world, no contradictions are true[[2]](#footnote-3).

The first problem is that, according to the theory as it was originally expressed by Priest (2005), if it is true that I think that the round square is round then it will be simultaneously false that I think that the round square is round; therefore, a contradiction will be true *in the actual world*. Of course, this consequence was not expected. In order to prevent it, Priest and Berto have suggested a solution that I will discuss and reject.

**2. The theory MM**

**2.1. Language**

I will outline the theory MM in very similar terms as Berto’s theory. (See Berto (2013) p.127.) The language of MM is a first-order language + a special operator ®. I will use the standard quantifiers ∃ and ∀, though Berto prefers to use Σ and Π. His notation may be misleading since the symbols he employs are commonly used to express substitutional quantification, yet in his theory they express objectual quantification. But the range of our quantifiers will be indeed wider than in standard logic since it will allow us to quantify over non-existent objects.

The special operator ® means *being represented*. If φ is a formula, then ®(φ) is a formula which reads: “it is represented that φ”.

Let us suppose that our primitive propositional operators are the negation (¬) and the conjunction (&); the others are defined as usual.

**2.2. Model**

A model of MM is a structure *M* = <𝓟, 𝓘, 𝓔, 𝓓, @, 𝓡, δ> satisfying the following conditions: 𝓟, 𝓘, 𝓔 and 𝓓 are disjoint sets. 𝓟 represents intuitively a set of possible worlds. 𝓘 and 𝓔 represent two different sets of impossible worlds. (Intuitively, the worlds in 𝓘 are only impossible because of truth-value gaps and gluts, while the worlds in 𝓔 are completely anarchic, every formula behaving as atomic.) Let us define the set of all worlds 𝓦 as the union of 𝓟, 𝓘 and 𝓔. 𝓓 is a non-empty set of individuals. @ is a member of 𝓟 and stands for the actual world. 𝓡 is a set of couples of members of 𝓦; intuitively, <*w*, *w*’> ∈ 𝓡 means that *w*’ is representationally accessible (or R-accessible) from *w*. (Note that this accessibility relation will not play any interesting role in what follows since we will only consider models with two worlds: the actual world @ and an impossible world *w* such that *w* is representationally accessible from @.)

The last term of a model M is a denotation function δ satisfying the following conditions (iδ)-(ivδ):

(iδ) For every individual constant *c*, δ(*c*) ∈ 𝓓.

(iiδ) For every couple <*w*, *Kn*> where *w* ∈ 𝓟 ∪ 𝓘, and *Kn* is an *n*-adic predicate,

δ(*w*, *Kn*) = <δV(*w*, *Kn*), δF(*w*, *Kn*)> where δV(*w*, *Kn*) and δF(*w*, *Kn*) are sets of *n*-tuples of members of 𝓓.

(We admit that the 1-tuple <*i*> is just *i*; thus, denotations of monadic predicates are sets of individuals.) Intuitively, δV(*w*, *Kn*) is the *positive extension* of *Kn* in *w*, i.e. the set of *n*-tuples of individuals of which *Kn* is *true* in *w*; wheras, δF(*w*, *Kn*) is the *negative extension* of *Kn* in *w*, i.e. the set of *n*-tuples of individuals of which *Kn* is *false* in *w*. (In standard logic, predicates only have one extension: the positive one; here, for a reason that will soon become clearer, predicates have this double extension: positive and negative.)

The third condition concerns positive and negative extensions in possible worlds in a possible world; positive and negative extensions are exhaustive and mutually exclusive:

(iiiδ) For every *w* ∈ 𝓟 and every *n*-adic predicate *Kn*:

δV(*w*, *Kn*) ∪ δF(*w*, *Kn*) = 𝓓*n* and δV(*w*, *Kn*) ⋂ δF(*w*, *Kn*) = ∅

Thus in a possible world, everything is either in the positive extension of *being round* or in its negative extension (but not both).

Concerning extensions of predicates, note that (iiδ) only concerns worlds in 𝓟 or 𝓘. We said nothing yet about extension of predicates in impossible worlds of the second kind, worlds of 𝓔. In those worlds, every formula will behave as a sort of predicate (it will allow logical failures such as formulas φ; ψ being true while their conjunction φ & ψ being not true). The general idea may seem simple; but, making it work is a little bit technical. First, we need to define the notion of the schema for a formula[[3]](#footnote-4). Let us say that the *free terms* of a formula are the individual constants and free variables occurring in that formula. (Thus, in ∃*x*(*Px* & *Pa* & *Py*), *a* and *y* are free terms but not *x*.) Now, for every formula φ of 𝓛𝓥 with *n* free terms (*n* may be 0), I will note [φ]*n* for the schema of φ: it is the expression resulting from substituting *t*1, …, *tn* to the *n* free terms in φ (in their order of appearance in the formula, from left to right). So, for example, the schema of *Pa* & *Pb* & *Qa* is the same as the schema of *Pb* & *Pa* & *Qb*, namely: [*Pt*1 & *Pt*2 & *Pt*1]2. The idea is to treat this schema as a sort of dyadic predicate in worlds of 𝓔, having positive and negative extensions. Thus, here is the fourth and last condition on δ:

(ivδ) For every couple <*w*, [φ]*n*> where *w* ∈ 𝓔 and [φ]*n* is an *n*-adic schema,

δ(*w*, [φ]*n*) = <δV(*w*, [φ]*n*), δF(*w*, [φ]*n*)> where δV(*w*, [φ]*n*) and δF(*w*, [φ]*n*) are sets of *n*-tuples of members of 𝓓.

Intuitively, δV(*w*, [*Pt*1 & *Pt*2]2) is the positive extension of [*Pt*1 & *Pt*2]2 in *w*. If the couple <*i*, *j*> belongs to this extension, and if δ(*a*) = *i* and δ(*b*) = *j*, then the formula *Pa* & *Pb* is true in *w*. (And it might be the case that, in *w*, neither *Pa* nor *Pb* is true.)

A careful reader may observe that there are formulas in which no free terms appear, for example: ∃*xPx*. Thus, there are 0-adic schemas. As it is not obvious to see how δ will work in this kind of case, I will come back to this issue later in 5.2.

**2.3. Semantic rules**

The semantic rules will define two semantic values: true and false. More precisely, they will define under which conditions a formula is true/false *in a model M at a world w for an assignation s*. The abbreviations of these two semantic values will be: *trueMws* and *falseMws*. It is very important to note that, although these two notions are defined almost completely independently (the only exception is rule (iiiv)-(iiif) for negation), every rule for a false formula mirrors the corresponding rule for a true formula and *vice versa*.

First, we need to define the notion of assignation. As usual, a function *s* is an assignation if, for every variable *v*, *s*(*v*) ∈ 𝓓. And we define the function δ*s*: if *t* is a constant then δ*s*(*t*) = δ(*t*); if *t* is a variable then δ*s*(*t*) = *s*(*t*).

Now we are ready to give the semantic rules[[4]](#footnote-5). Rules (iv)-(vf) define truth*Mws* and false*Mws* only for worlds *w* ∈ 𝓟 ∪ 𝓘. The last pair of rules (viv) and (vif) define truth*Mws* and false*Mws* for worlds *w* ∈ 𝓔.

(iv) *True atomic formula*

If *w* ∈ 𝓟 ∪ 𝓘 then *Knt*1…*tn* is true*Mws* iff <δ*s*(*t*1), …, δ*s*(*tn*)> ∈ δV(*w*, *Kn*).

(if) *False atomic formula*

If *w* ∈ 𝓟 ∪ 𝓘 then *Knt*1…*tn* is false*Mws* iff <δ*s*(*t*1), …, δ*s*(*tn*)> ∈ δF(*w*, *Kn*).

(iiv) *True quantified formula*

If *w* ∈ 𝓟 ∪ 𝓘 then ∃*v*(φ) is true*Mws* iff there is an assignation *s*’ that agrees with *s* except possibly at the variable *v* such that φ is true*Mws’*.

(iif) *False quantified formula*

If *w* ∈ 𝓟 ∪ 𝓘 then ∃*v*(φ) is false*Mws* iff for all assignation *s*’ that agrees with *s* except possibly at the variable *v*, φ is false*Mws’*.

(iiiv) *True negation*

If *w* ∈ 𝓟 ∪ 𝓘 then ¬φ is true*Mws* iff φ is false*Mws*.

(iiif) *False negation*

If *w* ∈ 𝓟 ∪ 𝓘 then ¬φ is false*Mws* iff φ is true*Mws*.

(ivv) *True conjunction*

If *w* ∈ 𝓟 ∪ 𝓘 then φ & ψ is true*Mws* iff φ is true*Mws* and ψ true*Mws*.

(ivf) *False conjunction*

If *w* ∈ 𝓟 ∪ 𝓘 then φ & ψ is false*Mws* iff φ is false*Mws* or ψ is false*Mws* (or both).

(vv) *True representational formula*

If *w* ∈ 𝓟 ∪ 𝓘 then ®(φ) is true*Mws* iff for every *w’* such that <*w*, *w*’> ∈ 𝓡, φ is true*Mw’s*.

(vf) *False representational formula*

If *w* ∈ 𝓟 ∪ 𝓘 then ®(φ) is false*Mws* iff for a *w’* such that <*w*, *w*’> ∈ 𝓡, φ is false*Mw’s*.

(viv) *True formula in worlds of* 𝓔

If *w* ∈ 𝓔 thenfor any formula φ whose schema is [φ]*n* and such that the terms *t*1, …, *tn* in [φ]*n* corresponds to the terms *t*1’, …, *tn*’ in φ, this formula φ is true*Mws* iff <δ*s*(*t*1’), …, δ*s*(*tn*’)> ∈ δV(*w*, [φ]*n*).

(vif) *False* *formula in worlds of* 𝓔

If *w* ∈ 𝓔 thenfor any formula φ whose schema is [φ]*n* and such that the terms *t*1, …, *tn* in [φ]*n* corresponds to the terms *t*1’, …, *tn*’ in φ, this formula φ is false*Mws* iff <δ*s*(*t*1’), …, δ*s*(*tn*’)> ∈ δF(*w*, [φ]*n*).

A formula is true*M* iff it is true*M@s* for every assignation *s* (in other terms, a formula is true in a model *M* iff it is true in *M* in the actual world for every assignation); and a formula is false*M* iff it is false*M@s* for every assignation *s*.

**3. From truth-value gluts and gaps in an impossible world**

**to truth-value gluts and gaps in the actual world**

Remember the special condition on the positive and negative extension of predicates in possible worlds:

(iiiδ) For every *w* ∈ 𝓟 and every *n*-adic predicate *Kn*:

δV(*w*, *Kn*) ∪ δF(*w*, *Kn*) = 𝓓*n* and δV(*w*, *Kn*) ⋂ δF(*w*, *Kn*) = ∅

The idea behind this condition is to make sure that, in possible worlds, every formula is either true or false (not both). Berto explicitly endorses this view in the following passage:

“Everything works familiarly enough as far as worlds in 𝓟 are concerned, the main change with respect to standard modal semantics being that truth and falsity conditions are spelt separately. But even this does not change much at possible worlds. The [condition (iiiδ)] dictates that, at each *w* ∈ 𝓟, any predicate *P* is either true or false of the relevant object (or n-tuples thereof), but not both. That no atomic formula is both true and false or neither true nor false entails that no formula is, as can be checked recursively. Overall, there are no so-called truth-value gluts or gaps at possible worlds.” (Berto 2013, p. 130).

Why distinguish positive and negative extensions? In possible worlds, a negative extension of a predicate *Kn* can be determined from its positive extension: it is its complement on 𝓓*n*. But the condition only holds for possible worlds; thus the purpose of this double extension is to allow truth-value gaps and gluts in *impossible worlds* of 𝓘. In those worlds, the restriction on positive and negative extensions does not hold.

Let us illustrate this point with an example. Consider the following model *M* = <𝓟, 𝓘, 𝓔, 𝓓, @, 𝓡, δ> with:

𝓟 = {@} 𝓘 = {*w*} 𝓔 = ∅ 𝓡 = {<@, *w*>}

𝓓 = {*i*} δ(*a*) = *i*

δV(@, *P*) = {*i*} δF(@, *P*) = ∅ δV(@, *Q*) = ∅ δF(@, *Q*) = {*i*}

δV(*w*, *P*) = {*i*} δF(*w*, *P*) = {*i*} δV(*w*, *Q*) = ∅ δF(*w*, *Q*) = ∅

One can easily check that (if we assume that the vocabulary of our language only contain *a*, *P* and *Q*) this structure, indeed, satisfies every condition spelled out in 2.2; consequently, it is a model of our theory. What does it say? In *M*, there is only one individual, which is denoted by *a*; in the actual world, *a* is *P* and is not *Q*. Therefore, in @, *Pa* is true and not false, and *Qa* is false and not true. There is no truth-value glut nor gap in @ for atomic formulas. But, in the impossible world *w*, the same individual *i* belongs to positive *and* negative extensions of *P*; yet, it doesn’t belong to either positive or negative extensions of *Q*. Therefore, in *w*, *Pa* is both true and false, and *Qa* is neither true nor false.

Let us first concentrate on the case of *Pa* in *w*. What about its negation, ¬*Pa*? According to rules (iiiv) and (iiif), ¬*Pa* is true in *w* iff *Pa* is false in *w*; *Pa* is indeed false[[5]](#footnote-6) in *w*, therefore ¬*Pa* is true in *w*. Additionally, ¬*Pa* is false in *w* iff *Pa* is true in *w*; *Pa* is indeed true in *w*, therefore ¬*Pa* is false in *w*. Conclusion: ¬*Pa* is both true and false in *w*.

According to rule (ivv) and (ivf), the conjunction *Pa* & ¬*Pa* will also be both true and false (it is true because *Pa* is true and ¬*Pa* is true; and false because *Pa* is false and ¬*Pa* is false.) So, there is a true (and false) contradiction in *w*, which is an impossible world; but in the actual world *Pa* & ¬*Pa* is just false and not true.

What about the representational formula ®(*Pa*)? Is it true in the actual world? According to (vv), ®(*Pa*) is true in the actual world iff in every world representationally accessible from the actual world it is true. The only world accessible from @ is *w*. *Pa* is true in *w*. Therefore, ®(*Pa*) is true in @.

Now, is ®(*Pa*) *false* in @? According to (vf), ®(*Pa*) is false in @ iff there is a world accessible from @ in which *Pa* is false. The world *w* is accessible from @ and *Pa* is false in *w*; therefore, ®(*Pa*) is false in @.

We can conclude that ®(*Pa*) is both true and false in @. And that also means that the contradiction ®(*Pa*) &¬®(*Pa*) is true and false in @. There is indeed a true contradiction in the actual world.

From the case of *Qa* which is neither true nor false in *w*, we could show something similar: the formula ®(*Qa*) is neither true nor false in the actual world. In order to be true, *Qa* should be true in *w*; and, to be false, *Qa* should be false in *w*. Therefore, the formula ®(*Qa*) ∨ ¬®(*Qa*) will neither be true nor false.

Finally, the condition (iiiδ) on positive and negative extensions of predicates in possible worlds is not enough to avoid truth value gluts and gaps in possible worlds. As soon as impossible worlds are represented in which formulas are neither true nor false or both true and false, the corresponding representational formulas in the actual world will also be neither true nor false or both true and false.

**4. A (not so good) solution**

There is an obvious solution to this problem (which has been mentioned by Priest and Berto): in the rule (vf), instead of requiring the represented formula to be false, we will only require it to be *not true*.

(vf\*) *False representational formula*

®(φ) is false*Mws* iff for a *w’* such that <*w*, *w*’> ∈ 𝓡, φ is not true*Mw’s*.

This rule solves the problem with ®(*Pa*) and ®(*Qa*) in M. Indeed, ®(*Pa*) becomes true and not false in @; and ®(*Qa*) becomes false and not true in @. In general, ®(φ) will be true and not false when φ is a truth-value glut in the represented world, and ®(φ) will be false and not true when φ is a truth-value gap in the represented world. Thus, (vf\*) prevents representation of truth-value gluts and gaps from contaminating the actual world (and possible worlds in general).

This modification is very important if one wants to apply the theory from a meinongian perspective: indeed, the purpose of the theory would be to represent worlds in which meinongian objects, such as the round square, have their properties. A meinongian theorist would obviously need worlds with truth value gluts and gaps. But if we keep (vf), the theory implies dialetheism, which is not expected. We need (vf\*) instead. So in books such as *Towards Non-Being* and *Existence as a Real Property* in which Priest and Berto defend meinongianism, we would expect them to clearly discuss this problem and its solution.

It is not the case. Berto only mentions the problem in footnotes (Berto (2013) p.130 footnote 4; Berto (2014) p.105 footnote 3) and Priest, as far as I know, says nothing about it in *Towards Non-Being*; however, something related to a similar problem can be found in a footnote in Priest (2008) (p.174, footnote 9).

It is disturbing to me that the correct formulation of one of the most important semantic rules of the theory is to be found only in footnotes. Moreover, according to Berto, this footnote is only “[a] technical note, which may be skipped without loss of continuity” (Berto (2013) p.130 footnote 4). I understand that footnotes are for technical details, but this modification of rule (vf) is not specially technical (there is nothing more technical in the formulation of (vf\*) than in the formulation of (vf)). And, most importantly, it is not a detail, especially from a meinongian point of view. Without this modification, the theory cannot work for impossible worlds with meinongian objects such as the round square, etc. Using (vf) instead of (vf\*) seems to defeat the purpose of the theory as a modal meinongianism.

And yet, to use (vf\*) would raise, in fact, a new and much deeper problem which I will explain. The rule (vf\*) has an interesting feature: a formula being false is defined in terms of a formula being true. Only one other rule has this feature: the rule (iiif) for the negation: ¬φ is false*Mws* iff φ is true*Mws*. But this rule has an equivalent for truth, the rule (iiiv), where a formula being true is defined in terms of a formula being false: ¬φ is true*Mws* iff φ is false*Mws*. More generally, we have observed that every rule for truth*Mws* corresponds *mutatis mutandis* to the rule for falsity*Mws*. Under this aspect, these two semantic notions seem correctly defined (it is obvious that truth and falsity are related notions and the semantic rules must express this relation). Turning our attention to the case of (vf\*), the corresponding rule for truth*Mws* is (vv); and while in (vf\*) a formula ®(φ) being false is defined in terms of φ being *not true*, in (vv) a formula ®(φ) being true is defined in terms of φ being *true*. Clearly, these two rules do not correspond to each other in the expected way. The rule (vf\*) is the only semantic rule which is not spelt out symmetrically from the corresponding rule for truth; therefore, though this rule (vf\*) solves our problem, it seems *ad hoc* and ill-defined for not corresponding properly.

Let us take an example. Suppose that we want to change our rule for true negation into this one: (iiiv\*) ¬φ is true*Mws* iff φ is not true*Mws*. Would it be acceptable not to change the corresponding rule for a false negation: (iiif) ¬φ is false*Mws* iff φ is true*Mws*? Negation would clearly be ill-defined; its truth table would then become:

|  |  |
| --- | --- |
| φ | ¬φ |
| T and F | F |
| T | F |
| F | T |
| neither T nor F | T |

According to this truth table, if φ is both true and false then ¬φ is just false and ¬¬φ is just true. It does not make any sense; this interpretation of ¬ is not consistent with the idea of negation.

So, if we change (iiiv) into (iiiv\*), we would expect to also change the corresponding rule for false negation in the following way: (iiif\*) ¬φ is false*Mws* iff φ is not false*Mws*. Then we would obtain an interpretation of negation with the following truth-table:

|  |  |
| --- | --- |
| φ | ¬φ |
| T and F | neither T nor F |
| T | F |
| F | T |
| neither T nor F | T and F |

In this interpretation of negation, the negation of a truth-value gap is a truth value glut and *vice versa*. It is somehow unnatural; but, the problem is not purely logical. We can see that this operation behaves indeed as a sort of negation, though it is not in an intuitive way (we might say that it is a *radical negation* that inverts every truth-value, even gluts and gaps; so, we have at least this feature of negation: for any formula φ the semantic value of ¬¬φ is the semantic value of φ.) It gives us an odd interpretation of negation, but it is a consistent interpretation.

The point here is not about negation but about how to modify a pair of semantic rules for true and false formulas in a consistent way: we must modify both rules, not just one. What seems *ad hoc* and ill-defined in (vf\*) is that we have not modified the other rule (vv) in the same way. Would it solve our problem? Let us try. If we change (vf) into (vf\*), we should also change (vv) into this rule:

(vv\*) *True representational formula*

®(φ) is true*Mws* iff for every *w’* such that <*w*, *w*’> ∈ 𝓡, φ is not false*Mw’s*.

Now, the rules (vv\*) and (vf\*) seem to correspond to each other in a consistent way, but we fall into contradiction again. According to these rules, ®(*Pa*) would be neither true nor false in @; additionally, ®(*Qa*) would be both true and false in @ (the representation of a truth value glut becomes a truth value gap and *vice versa*; the result is just worse).

From a more general perspective, we could ask: what is the purpose of defining positive and negative extensions for predicates and giving two separate sets of semantic rules for truth and falsity? The idea is that truth and falsity are supposed to be on a foot of equality. But with (vf\*), it becomes clear that truth has a special status; it is *the* semantic value. (Otherwise, it would be as acceptable to keep (vf) and modify (vv) into (vv\*). But this solution seems absurd: it would make ®(*Pa*) false and not true in @, and ®(*Qa*) true and not false in @).

There is another difficulty concerning (vf\*). Suppose that in a world *w* accessible from @, the formula φ is both true and false while the formula ψ is just true (and not false). We should expect their respective representations to be different; but how will they differ? If we keep rules (vv) and (vf) (and thus accept truth-value gluts and gaps in the actual world), it is straightforward: ®(φ) is true and not false while ®(ψ) is both true and false. The representational formula just has the semantic values of the represented formula. But with (vf\*), the formulas ®(φ) and ®(ψ) will come out true (just true, not false). What will distinguish them is the status of the representations of their negations: ®(¬φ) will be true but not ®(¬ψ). Similarly, what will distinguish the representation of a formula φ that is false and not true and the representation of a formula ψ that is neither true nor false? For both of them, ®(φ) and ®(ψ) will be just false; but ®(¬φ) will be true while ®(¬ψ) will be false. In conclusion, the fact that a formula is a truth-value gap or glut in the represented world can only be represented through the representation of the negation of this formula.

This feature suggests the following modification to the theory: get rid of negative extensions and semantic rules for falsity (just keep normal (positive) extension and semantic rules for truth, which is standard); then, define impossible worlds of 𝓘 as worlds where the negations of atomic formula behave like atomic formulas. This solution is, in fact, very much in line with Priest’s account of impossible worlds of 𝓔, where every *n*-adic schema of formula behave like an *n*-adic predicate. So, such a solution would be very consistent with the rest of the semantics. It would give a unified account of the different sort of impossible worlds. Impossible worlds, in general, are simply worlds where non-atomic formulas behave like atomic ones (we could say of those formulas that they are *pseudo-atomic formulas* in those worlds. The simplest kind of impossible worlds, which corresponds to the worlds of 𝓘, are the worlds in which the only pseudo-atomic formulas are the negations of atomic formulas).

My conclusion is that the problem with (vf) raises the dilemma that we either accept (vf) and its unpleasant consequences (there are true contradictions in the actual world) or we have to reject (vf), but then we have good reasons to reject more generally Priest’s semantics with positive and negative extensions for predicate and separate sets of rules for truth and falsity.

In the following section, I will point out two other minor issues in MM. And in the last section, I will introduce a simpler version of modal meinongianism whose expressive power is at least equivalent to Priest’s and Berto’s account.

**5. Two other issues in MM**

**5.1. What if nothing is represented?**

There is another unexpected consequence of rules (vv) and (vf). Suppose that no world is R-accessible (*i.e.* representationally accessible) from the actual world. According to rule (vv), a formula ®(φ) is true in @ iff, in every world R-accessible from @, φ is true. Since no world is R-accessible from @, this condition will be trivially satisfied; therefore, for any formula φ, it will be true that ®(φ).

Similarly, according to (vf), no formula of the form ®(φ) will be false since there is no R-accessible world in which φ is false (even with the modified rule (vf\*) it would be the same).

Thus, in @, every formula ®(φ) will be true and not false. It sounds particularly absurd since such a model seems to state precisely a situation where nothing is represented in the actual world; we would rather expect every formula ®(φ) to be false and not true.

This problem has a straightforward solution (and I think Priest and Berto would accept it). In (vv), we must add a clause concerning the fact that there must be at least one R-accessible world; and in (vf) we must add a clause concerning the fact that the representational formula is false if there is no R-accessible world. Thus, the correct rules should be:

(vv#) *True representational formula*

If *w* ∈ 𝓟 ∪ 𝓘 then ®(φ) is true*Mws* iff there is a *w*’such that <*w*, *w*’> ∈ 𝓡, and for every *w’* such that <*w*, *w*’> ∈ 𝓡, φ is true*Mw’s*.

(vf#) *False representational formula*

If *w* ∈ 𝓟 ∪ 𝓘 then ®(φ) is false*Mws* iff there no *w*’such that <*w*, *w*’> ∈ 𝓡, or for a *w’* such that <*w*, *w*’> ∈ 𝓡, φ is false*Mw’s*.

According to these rules, if no world is R-accessible from a possible world *w*, then every formula ®(φ) will be false and not true in *w*.

**5.2. Something about 0-adic schema**

In worlds of 𝓔, *n*-adic schemas behave like *n*-adic predicates. It is quite easy to understand how it works for *n* ≥ 1. But what about 0-adic schemas?

For any formula, you obtain its schema by replacing its free terms (i.e. individual constants and free variables) by *t*1, *t*2, etc. For example, for the formula ∃*x*(*Px* & *Pa* & *Py*), you can construct the dyadic schema: [∃*x*(*Px* & *Pt*1 *Pt*2]2. But what is the schema of ∃*xPx*? There is no free term in this formula; so its schema will be: [∃*xPx*]0, a 0-adic schema.

How are those formulas supposed to behave in worlds of 𝓔? The general idea is that, in these worlds, each formula behaves as atomic. Therefore, there should be worlds of 𝓔 in which ∃*xPx* is true and other worlds where it is not true (and it should be completely independent from the truth-values of other formulas).

Let us look at this in detail. Here is the fourth condition on denotation function:

(ivδ) For every couple <*w*, [φ]*n*> where *w* ∈ 𝓔 and [φ]*n* is an *n*-adic schema,

δ(*w*, [φ]*n*) = <δV(*w*, [φ]*n*), δF(*w*, [φ]*n*)> where δV(*w*, [φ]*n*) and δF(*w*, [φ]*n*) are sets of *n*-tuples of members of 𝓓.

This rule works fine for *n* ≥ 1. For example, the positive extension of a dyadic schema [*Pt*1 & *Pt*2]2 in a world *w* of 𝓔 will be a set of couples of members of 𝓓, like the extension of a dyadic predicate. But how can we read this rule for 0-adic schema? The positive extension of a 0-adic schema in *w* ∈ 𝓔 should be a set of 0-tuples of members of 𝓓. There is only one 0-tuple: the empty set. Therefore, the positive extension of a 0-adic schema is either {∅} (the extension contains this unique 0-tuple) or ∅ (the extension is just empty). Let us define: {∅} = 1 and ∅ = 0. We can say that the positive extension of a 0-adic schema is either 1 or 0 (and we can say the same for its negative extension).

Now, here are the semantic rules for formulas in worlds of 𝓔:

(viv) *True formula in worlds of* 𝓔

If *w* ∈ 𝓔 thenfor any formula φ whose schema is [φ]*n* and such that the terms *t*1, …, *tn* in [φ]*n* corresponds to the terms *t*1’, …, *tn*’ in φ, this formula φ is true*Mws* iff <δ*s*(*t*1’), …, δ*s*(*tn*’)> ∈ δV(*w*, [φ]*n*).

(vif) *False* *formula in worlds of* 𝓔

If *w* ∈ 𝓔 thenfor any formula φ whose schema is [φ]*n* and such that the terms *t*1, …, *tn* in [φ]*n* corresponds to the terms *t*1’, …, *tn*’ in φ, this formula φ is false*Mws* iff <δ*s*(*t*1’), …, δ*s*(*tn*’)> ∈ δF(*w*, [φ]*n*).

For the 0-adic case, these rules state that φ is true*Mws* iff ∅ ∈ δV[φ]*n*(*w*). In other terms, φ is true*Mws* iff δV[φ]*n*(*w*) = 1. And similarly, φ is false*Mws* iff δF[φ]*n*(*w*) = 1.

Everything seems to work fine: 0-adic schemas will indeed behave as atomic formulas. So, where is the problem?

First, there is a technical problem concerning 0-adic schemas such as ∃*xPx* and ∃*yPy*. Since they are distinct 0-adic schemas, it is possible for ∃*xPx* to be true in a world and ∃*yPy* to be not true in the very same world. It does not seem to make any sense. However, we will see soon that a technical solution can be easily provided.

There is something more to say about 0-adic schemas: looking at the way the theory works for 0-adic schemas in worlds of 𝓔 gives us, I think, good reasons to rethink the way it could work more generally in those worlds. Let me explain why.

Somehow, everything works as if, in worlds of 𝓔, the formulas corresponding to 0-adic schemas (namely formulas without free terms) denote one of those four couples: <1, 1>, <1, 0>, <0, 1>, <0, 0>. When a formula denotes <1, 1>, it means that it is both true and false; when it denotes <1, 0>, it means that it is true and not false. Etc. To say it in simpler terms: everything works as if, in worlds of 𝓔, those formulas denote directly truth-values (and it is consistent with the idea that formulas, in those impossible worlds, behave as atomic).

In MM, this outcome is merely a side-effect of (ivδ), (viv) and (vif), and it concerns only formulas without free terms. But now that we see how it works here, do we have any good reason for not doing the same for every formula?

Doing the same for formulas without variables is easily applicable: they could perfectly denote an arbitrary truth-value. But doing the same for formulas with variables would be a bad idea[[6]](#footnote-7). *Px* could be true in a world but not *Py*, which is absurd. Similarly, ∃*xPx* could be true in a world but not ∃*yPy*. The simplest solution would be, here again, to use a schema of formula: but a schema for variables only.

Let us say that the *v*-schema of φ is the expression resulting from substituting meta-variables *v*1, …, *vn* to the *n* variables in φ (in their order of appearance in the formula, from left to right). The *v*-schema of a formula without variables is itself. Thus, the *v*-schema of ∃*y*(*Px* &*Py*) and the *v*-schema of ∃*x*(*Py* & *Px*) are the same: ∃*v*1(*Pv*2 & *Pv*1). Let us use the notation |φ| for the *v*-schema of φ.

As such, instead of (ivδ) we could have the following condition:

(ivδ\*) For every couple <*w*, φ> where *w* ∈ 𝓔 and φ is a formula, δ(*w*, |φ|) is a couple of members of {1, 0}.

And the rules (viv) and (vif) would be:

(viv\*) *True formula in worlds of* 𝓔

If *w* ∈ 𝓔 then φ is true*Mws* iff the first member of δ(*w*, |φ|) is 1.

(vif\*)*False formula in worlds of* 𝓔

If *w* ∈ 𝓔 then φ is false*Mws* iff the second member of δ(*w*, |φ|) is 1.

The resulting semantics is much simpler. It will also be different under certain aspects. Suppose that in a model *M* the same individual *i* is denoted by two individual constants *a* and *b*. In the original version of MM, if in an impossible world *w* ∈ 𝓔, the positive extension of the monadic schema [*Pt*1]*n* contains only *i*, then *Pa* and *Pb* will be automatically true in *w*. More generally, *Pa* will be true in a world iff *Pb* is also true. Consequently, according to the original version of the theory, even in impossible worlds, there is no case where a formula about *a* is true and the same formula about *b* is not: co-referential terms are substitutable *salva veritate* in every context. It seems to be a very odd limitation of impossible worlds: there are worlds where *Pa* and *Qa* are true but not *Pa* & *Qa*, yet there is no world where *Pa* is true and not *Pb* if *a* and *b* denote the same thing. Is it *too impossible* even for MM?

On the contrary, in the modified version of MM (with (ivδ\*), (viv\*) and (vif\*), this restriction does not hold. Even if *a* and *b* denote the same individual, there could be an impossible world *w* ∈ 𝓔 in which *Pa* is true and *Pb* is not: it only requires that δ(*w*, |*Pa*|) = <1, 0> and δ(*w*, *Pb*) = <0, 1>. It seems to me that the theory looks better in this way: it explains why substitutivity of identical terms fails in worlds of 𝓔.

I will follow this idea in the last section of this paper for constructing a revised and simpler version of modal meinongianism.

**6.Simple modal meinongianism**

The theory I introduce in this section is much simpler than MM; predicates will only have positive extension, there will be semantic rules for truth only, I will not distinguish different sorts of worlds (except the actual world), and rules for interpreting formulas in non-actual worlds will be very simple. I must mention, however, that I had to slightly complicate the semantic rule for negation; but, it seems at an acceptable cost (see 6.3.) My theory will be able to do the same kind of work as MM; in particular, I will show how the most important notions of MM can be defined in this new theory without any loss.

**6.1. Models**

Language is the same as before. A model is a structure *M* = {𝓦, 𝓓, @, 𝓡, δ} satisfying the following conditions. 𝓦 and 𝓓 are disjoint sets. Intuitively, 𝓦 is the set of worlds and @, the actual world, is one of its members. Let us call 𝓦\* the set of non-actual worlds (the set 𝓦 without @). 𝓓 is, as usual, the non-empty set of individuals. 𝓡 is, as in MM, a set of couple of members of 𝓦. (<*w*, *w*’> ∈ 𝓡 means that *w*’ is R-accessible from *w*.) And finally δ is a denotation function satisfying these three conditions (iδ)-(iiiδ):

(iδ) For every individual constant *c*, δ(*c*) ∈ 𝓓.

(iiδ) For every *n*-adic predicate *Kn*, δ(*Kn*) ∈ 𝓓*n*.

(Denotation of predicates is not relativized to worlds. Intuitively, δ(*Kn*) is the extension of *Kn* in the actual world. In other worlds, predicates will also receive a sort of extension, or pseudo-extension, but it will be defined later; it is not a primitive notion of the theory.)

The third condition on δ is the following:

(iiiδ) For certain[[7]](#footnote-8)couples <*w*, |φ|> where *w* ∈ 𝓦\* and φ is a formula, δ(*w*, |φ|) ∈ {1, 0}.

Remember that |φ| is the *v*-schema of the formula, as defined in section 5.2.

Intuitively, δ(*w*, |φ|) = 1 means that φ is true in *w*, and δ(*w*, |φ|) = 0 means that φ is not true in *w*. Note that δ(*w*, |φ|) is not always defined: it is defined for *certain* couples <*w*, φ>; if it is defined, I will call φ a pseudo-atomic formula of *w*. There are pseudo-atomic formulas only in non-actual worlds (because we require the world *w* to be a member of 𝓦\*, not 𝓦).

**6.2. Semantic rules**

As usual, a function *s* is an assignation if, for every variable *v*, *s*(*v*) ∈ 𝓓. We define the function δ*s* as before: if *t* is a constant then δ*s*(*t*) = δ(*t*); if *t* is a variable then δ*s*(*t*) = *s*(*t*).

The semantic rules will define the notion of truth*Mws* (i.e. truth in a model *M* in a world *w* for an assignation *s*). The first rule (O) is a special “overriding” rule that specifies under which conditions a pseudo-atomic formula is true or not true in a world. This rule overrides the other regular rules: the idea is that, if we want to determine if a formula φ is true or not true in a certain world *w*, we must first check the value of δ(*w*, |φ|). If it is defined, then the formula is true in *w* if the value is 1 and not true if the value is 0, and there is nothing more to say. If δ(*w*, |φ|) is not defined, then, and only then, we must follow the standard semantic procedure with rules (i) to (v).

Here are the six semantic rules of our theory:

(O) *Pseudo-atomic formula*

If δ(*w*, |φ|) is defined, then φ is true*Mws* iff δ(*w*, |φ|) = 1.

If δ(*w*, |φ|) is not defined, then (and only then) the following rules (i)-(v) apply.

(i) *Atomic formula*

*Knt*1…*tn* is true*Mws* iff *w* is @ and <δ(*t*1), …, δ(*tn*)> ∈ δ(*Kn*).

(ii) *Quantified formula*

∃*v*(φ) is true*Mws* iff there is an assignation *s*’ that agrees with *s* except possibly at the variable *v* such that φ is true*Mws’*, or a formula φ(*t*/*v*) (i.e. the result of substituting the free occurrences of *v* in φ by a term *t*) is true*Mws*[[8]](#footnote-9).

(iii) *Negation.* ¬φ is true*Mws* iff φ is not true*Mws*.

(iv) *Conjunction.* φ & ψ is true*Mws* iff φ is true*Mws* and ψ is true*Mws*.

(v) *Representational formula*.

®(φ) is true*Mws* iff there is at least one *w’* such that <*w*, *w*’> ∈ 𝓡, and for every *w’* such that <*w*, *w*’> ∈ 𝓡, φ is true*Mw’s*.

We define truth*M* (truth in a model *M*) as truth*M@s* for every assignation *s*. Let us call this theory SMM.

**6.3. A complication with negation**

I think that SMM (or a close variation of it, augmented with identity for example) can do everything MM was intended to do. I will show how certain notions that seem to be lost in SMM (for example positive and negative extensions of predicates, and extensions at non-actual worlds) can be defined.

As a first exercise, let us build a model reproducing the model from section 3. Intuitively, it is a model in which there is only one individual denoted by *a*; in the actual world, *a* is *P* and is not *Q*; but in a represented world, *a* is both *P* and not *P* and neither *Q* nor not *Q*. We can express it with the model *M* = {𝓦, 𝓓, @, 𝓡, δ} with:

𝓦 = {@, *w*} 𝓓 = {*i*} 𝓡 = {<@, *w*>}

δ(*a*) = *i* δ(*P*) = {*i*} δ(*Q*) = ∅

δ(*w*, |*Pa*|) = 1 δ(*w*, |¬*Pa*|) = 1 δ(*w*, |*Qa*|) = 0 δ(*w*, |¬*Qa*|) = 0

In @, the formula ®(*Pa* & ¬*Pa*) is true and the formula ®(*Qa* ∨ ¬*Qa*) is not true. In *w*, we could say that *Pa* is a truth-value glut because both *Pa* and its negation are true, and *Qa* is a truth-value gap because both *Qa* and its negation are not true. It might seem that everything works fine but, in fact, there is something unsatisfying. Intuitively, if *Pa* is a glut, then so should be ¬*Pa* and ¬¬*Pa*, etc.; that is to say ¬*Pa*, ¬¬*Pa*, etc., should be all true. But, in this model, ¬¬*Pa* is not true. Indeed, δ(*w*, |¬¬*Pa*|) is not defined, therefore the truth-value of ¬¬*Pa* is only determined by the rule (iii), according to which ¬¬*Pa* is true in *w* iff ¬*Pa* is not true in *w*; since ¬*Pa* is true, ¬¬*Pa* is not true.

The formula *Qa* raises a similar issue: *Qa* is a gap because neither *Qa* nor ¬*Qa* is true; but ¬¬*Qa* is true.

How can we deal with this issue? In fact, there is something wrong with the interpretation of negation and we should modify (iii). Negation should be defined in such a way that the negation of a truth-value glut like *Pa* is also a truth-value glut, and such that the negation of a truth-value gap like *Qa* is also a truth-value gap. But our first problem is that we do not have a clear understanding of what is a truth-value glut or gap in our semantics. Intuitively, a glut is a formula both true and false, and a gap is a formula neither true nor false; but in SMM, so far, I have only introduced the notion of truth. Why *Pa* seems to be a truth-value glut in *w*? Because *Pa* and its negation are both true in *w*. And why *Qa* seems to be a truth-value gap in *w*? Because neither *Qa* nor its negation is true. Thus, if we want them to be respectively a glut and a gap, we could define falsity from truth in the following way:

(*f*) A formula φ is false*Mws* iff its ¬φ is true*Mws*.

(Truth*Mws* is still the only *primitive* semantic value of SMM; falsity*Mws* is just a defined notion. It is very different from MM where truth*Mws* and falsity*Mws* were both primitive semantic values.)

A formula being false means that its negation is true. It is easy to see that, according to this definition, *Pa* in *w* is indeed both true and false and *Qa* is neither true and false (but ¬*Qa* is false since its negation ¬¬*Qa* is true).

Now that we have made clear what *true* and *false* means, we can express how our operator ¬ is expected to behave. We want to use this truth-table[[9]](#footnote-10):

|  |  |
| --- | --- |
| φ | ¬φ |
| T and F | T and F |
| T | F |
| F | T |
| neither T nor F | neither T nor F |

Negation, as it is defined in (iii), does not work in this way in every world. We have seen for example that *Pa* is true and false, but ¬*Pa* is just true (because ¬¬*Pa* is not true). We can solve this particular issue by requiring that a double negation ¬¬φ is true*Mws* iff φ is true*Mws*. If we add this special clause in (iii) we solve *this* problem, but every other operator of our language will raise a problem of its own. For example, conjunction: here we expect *Pa* & *Pa* to be a truth-value glut as well as *Pa*. However, according to our rule for negation, ¬(*Pa* & *Pa*) is not true, and thus *Pa* & *Pa* is just true and not false. So we would also need a special clause for conjunctions.

Let us go straight to the solution. There is a semantic rule for negation that makes everything work fine; the rule introduces a set of sub-conditions (O¬)-(v¬) that follows the structure of semantic rules (O)-(v). Basically, the rule states a different condition for a negation ¬φ to be true in *w*, depending on the form of φ: the formula φ can be an atomic formula, a quantified formula, a negation, a conjunction, a representational formula, and nothing else. For each case, the condition for the truth of ¬φ in *w* will be different. For example, if φ is a negation of form ¬ψ, we expect ψ to be true in *w* (but only if δ(*w*, φ) is not defined, otherwise this condition overrides the other rules).

Here is the complete rule:

(iii\*) *Negation*

¬φ is true*M*@*s* iff φ is not true*M*@*s*.

For *w* ∈ 𝓦\*:

(O¬) if δ(*w*, |φ|) is defined, then ¬φ is true*Mws* iff δ(*w*, |φ|) = 0.

if δ(*w*, |φ|) is not defined, then the following sub-rules (ii¬)-(v¬) apply[[10]](#footnote-11):

(ii¬) if φ is a quantified formula ∃*v*(ψ), then ¬φ is true*Mws* iff for any assignation *s*’ that agrees with *s* except possibly at the variable *v*, ψ is false*Mws’*, and every formula φ(*t*/*v*) is false*Mws*.

(iii¬) if φ is a negation ¬ψ, then ¬φ is true*Mws* iff ψ is true*Mws*.

(iv¬) if φ is a conjunction ψ & ψ’, then ¬φ is true*Mws* iff ψ is false*Mws* or ψ’ is false*Mws*.

(v¬) if φ is a representational formula ®(ψ), then ¬φ is true iff there is no *w*’ such that <*w*, *w*’> ∈ 𝓡, or there is *w*’ such that <*w*, *w*’> ∈ 𝓡 and the formula ψ is false*Mw*’*s*.

With this rule, negation will behave as expected. In particular, one can easily see that not only *Pa* but also ¬¬*Pa* and *Pa* & *Pa* will be truth-value gluts in *w*.

One might be concerned that SMM is less simple than imagined. I admit there is indeed something complex in this interpretation of negation. But there is an advantageous feature of SMM with its modified rule for negation: I have defined a full-fledged four-valued logic in a classical framework; the only primitive semantic value is still the truth*Mws*. More generally, looking at the primitive notions of the theory, SMM remains much simpler than MM.

**6.4. Defining notions**

We have already defined in SMM a central notion of MM: falsity*Mws*. Now I will show how we can define in SMM various other notions of MM. First, let us classify different sorts of worlds in the way we did in MM. Remember that, in SMM, a formula φ is a pseudo-atomic formula of a world *w* iff δ(*w*, |φ|) is defined. Let us define 𝓟 and 𝓘 as follow:

𝓟 is the set of worlds *w* such that φ is a pseudo-atomic formula of *w*, iff φ is an atomic formula.

𝓘 is the set of worlds *w* such that φ is a pseudo-atomic formula of *w*, iff φ is an atomic formula or a negation of an atomic formula.

In worlds of 𝓟, atomic formulas receive arbitrary truth-values, then everything else follows rules (ii)-(v). In worlds of 𝓘, not only atomic formulas but also negations of atomic formula receive arbitrary truth-values, then everything else follows rule (ii)-(v). It is easy to see that in worlds of 𝓟, every formula will be either true or false but not both, while in worlds of 𝓘, there can be truth-value gluts and gaps (in our model *M* the world *w* was a member of 𝓘). However, logics for connectors and quantifiers behave consistently in worlds of 𝓘.

The impossible worlds 𝓔 would simply be the complement of 𝓟 ∪ 𝓘 on 𝓦. In these worlds, any formula could be a pseudo-atomic formula.

Let us finally define the notions of positive and negative extensions: if *w* is a world of 𝓟 or 𝓘, and *Kn* is an *n*-adic predicate, then the positive extension of *Kn* in *w* is the set of *n*-tuples <*t*1, …, *tn*> such that *Kt*1…*tn* is true*Mws*, and its negative extension is the set of *n*-tuples <*t*1, …, *tn*> such that *Kt*1…*tn* is false*Mws*.

**7. Conclusion**

In conclusion, although the primitive notions of SMM are much simpler and standard (just one primitive semantic value: truth*Mws*; predicates only have one extension), SMM seems as powerful as MM: every notion of MM (at least every important notion) is definable in SMM. If a theory can do the same work as MM for a lesser cost, this seems to be a good reason to prefer it. The only cost in SMM comes from the interpretation of negation in non-actual worlds.

In any case, if Priest and Berto want to support their conception of modal meinongianism (with positive and negative extensions and separate sets of rules for truth and falsity), they should clarify how to deal with semantic rules for representational formulas and address the critical points I raise in section 4.

Bellnap, Nuel, 1977, “A useful four-valued logic”, in: J.M. Dunn and G. Epstein (eds.), Modern Uses of Multiple-Valued Logic, Dordrecht: D. Reidel Publishing Co., 8–37.

Berto, Francesco, 2008, “Modal Meinongianism for Fictional Objects”,*Metaphysica*, 9, 2: 205-218.

- 2011, “Modal Meinongianism and Fiction: the Best of Three Worlds”, *Philosophical Studies*, 152, 3: 313-334.

- 2013, *Existence as a Real Property*, Synthese Library, Springer Netherlands.

- 2014, ‘‘On Conceiving the Inconsistent’’, *Proceedings of the Aristotelian Society*, 114: 102-121.

Dunn, J. Michael, 1976, “Intuitive semantics for first-degree entailment and ‘coupled trees’”, *Philosophical Studies*, 29: 149–168.

Parsons, Terence, 1980, *Nonexistent Objects*, New Haven: Yale University Press.

Priest, Graham, 2005, Towards Non-Being. The Logic and Metaphysics of Intentionality, Oxford: Clarendon.

- 2008, *An Introduction to Non-Classical Logic*, 2nd edition, New York: Cambridge University Press.

Zalta, Edward N., 1983, *Abstract objects: An Introduction to Axiomatic Metaphysics*, Dordrecht: D. Reidel

1. I would like to thank an anonymous reviewer for pointing out to me the relevant footnotes in Priest and Berto’s work. [↑](#footnote-ref-2)
2. See Priest 2005 p.20-21. Though Priest is sympathetic to a dialetheist view, he makes clear in this passage that according to the standard version of his theory there are no truth-value gaps nor gluts in the actual world. [↑](#footnote-ref-3)
3. My technique differs slightly from Priest’s but it does the same work. [↑](#footnote-ref-4)
4. Those are the rules as they are given (*mutatis mutandis*) by Priest (2005) and by Berto (2013); but a footnote in Berto (2013) will bring a significant modification to be discussed later. [↑](#footnote-ref-5)
5. It would be a mistake to think that *Pa* is not false for the reason that it is *true and false*; being *true and false* is not a special semantic value; it is nothing but *being true* and *being false*. Therefore, *Pa* is indeed false in *w* (which does not exclude that it is also true). [↑](#footnote-ref-6)
6. I thank an anonymous reviewer for pointing that out. [↑](#footnote-ref-7)
7. By “certain” I mean a random number: it might be every couple <*w*, φ>, only some of them but not all, or even none of them. [↑](#footnote-ref-8)
8. We need this second clause for the following reason: if δ(*w*, |*Pa*|) = 1, then *Pa* is true*Mws* and we expect ∃*x*(*Px*) also to be true*Mws*. But the first clause will not do the work: *Pa* is not true*Mws* because of *a* denoting something that *x* might also denote; *Pa* is true*Mws* only because it is assigned to be true*Mws*. Thus, if ∃*xPx* is true*Mws* it will only be because of a formula of form *Pt* being true*Mws*. [↑](#footnote-ref-9)
9. It corresponds to Dunn and Belnap’s interpretation of negation in their four-valued logic. (See Dunn 1976, Belnap 1977). It seems indeed to be the most natural way to understand negation in a four-valued semantics. [↑](#footnote-ref-10)
10. There is no sub-rule (i¬) for atomic formula because if φ is an atomic formula and δ(*w*, |φ|) is not defined, then ¬φ is always not true. (To say it in other terms: a negation of an atomic formula φ is true in a non-actual world *w* only if δ(*w*, |φ|) = 0.) [↑](#footnote-ref-11)