

Christopher Piñón (Budapest)*

Negating right boundary achievements (Comments on Malink)

1. Introduction

In this paper, Malink's (2007) account of right boundary achievements under conative negation is critically reviewed. Two potential problems are described for it, one concerning the question whether presuppositions are modifiable, and another concerning the usefulness of the kind of presupposition that Malink assumes for right boundary achievements. Finally, an alternative account is proposed that succeeds in avoiding these two potential issues.

2. Malink's proposal

Malink (2007) proposes an account of right boundary achievements under 'conative negation'. Basically, a right boundary achievement is an achievement that denotes a natural right boundary of an activity, where 'natural right boundary' means that an activity of the type in question cannot continue after the boundary has been reached.¹ Two examples of right boundary achievements in German are as follows:

- (1) a. Peter fand den Schlüssel.
Peter found the key
'Peter found the key.'
- b. Peter gewann das Spiel.
Peter won the game
'Peter won the game.'

A further idea in this connection is that the activity that a right boundary achievement denotes the right boundary of is lexically presupposed. Support for this idea

* This is a significantly revised version of a commentary that I presented on 25 June 2005 at *Workshop: Satz und Kontext*, Leipzig, 25–26 June 2005. I would like to thank Marko Malink, Ingolf Max, and Anita Steube for inviting me to be a commentator at the workshop. I am grateful to Marko and Fabienne Martin for a number of fruitful exchanges about this topic. This work was supported by the Hungarian Scientific Research Fund (OTKA TS 049873).

¹ An account of achievements in terms of boundaries is presented in Piñón (1997). See also Heyde-Zybatow (2004) for a discussion of right boundary achievements.

comes from the observation that the existence of such an activity is preserved under negation:

- (2) a. Peter fand den Schlüssel nicht.
 Peter found the key NEG
 ‘Peter didn’t [couldn’t] find the key.’
 b. Peter gewann das Spiel nicht. (= Malink’s (6a))
 Peter won the game NEG
 ‘Peter didn’t [couldn’t] win the game.’

These negative sentences, no less than the corresponding positive ones in (1), imply that Peter searched for the key and that he played the game, respectively.²

Malink points out that the negation of a right boundary achievement in the present tense with the unmarked prosody (= stress on the direct object) imply that something like the presupposed activity takes place at the speech time:

- (3) a. Peter findet den *Schlüssel* nicht. (= Malink’s (7a))
 Peter finds the key NEG
 ‘Peter can’t find (isn’t finding) the key.’
 b. Peter gewinnt das *Spiel* nicht. (= Malink’s (7c))
 Peter wins the game NEG
 ‘Peter can’t win (isn’t winning) the game.’

In (3a), for example, we have the impression that Peter is searching for or at least making an effort to find the key at the speech time. Malink labels the negation in (3) ‘conative negation’ and suggests that conative negation induces an aspectual shift by turning an achievement into an activity, which can then receive a proper present tense reading.

At the same time, he observes that negation does not always induce this kind of aspectual shift with right boundary achievements. In particular, negation does not effect this kind of aspectual shift with the so-called I-topic prosody:

- (4) a. *Peter* findet den Schlüssel *nicht*. (= Malink’s (8a))
 Peter finds the key NEG
 ‘Peter won’t find the key.’
 b. *Peter* gewinnt das Spiel *nicht*. (= Malink’s (8c))
 Peter wins the game NEG
 ‘Peter won’t win the game.’

² There is another sense of *finden* for ‘accidental findings’ that does not presuppose a searching activity per se. This sense figures most prominently in the case of an indefinite object, e.g., *Peter fand einen Schlüssel* ‘Peter found a key’, which does not presuppose that Peter had been searching for a key. However, for expediency I set aside this use of *finden* here.

He claims (preceding his (8)) that these sentences receive a ‘prospective negation reading’ which denies the existence of the achievement event in the future without saying anything about the existence of a preceding activity at or after the speech time.

Malink also provides crosslinguistic evidence from right boundary achievements in Czech and Ancient Greek in support of conative negation as an inducer of the aspectual shift to an activity. To take one of his examples from Czech, observe that whereas the negation of the imperfective verb *nacházet* ‘find.IMPF’ in the present tense is an instance of conative negation, as in (5a) (cf. (3a)), the negation of the corresponding perfective verb *najít* ‘find.PF’ is not, as in (5b) (cf. (4a)).

- (5) a. Petr nenachází svůj klíč. (= Malink’s (11a))
 Petr NEG.finds.IMPF his key
 ‘Petr can’t find (isn’t finding) his key.’
 b. Petr svůj klíč nenajde. (= Malink’s (12a))
 Petr his key NEG.finds.PF
 ‘Petr won’t find his key.’

As he notes, the negation in (5a) cannot be the usual negation of the corresponding positive sentence, because the latter cannot receive an episodic reading in the present tense:³

- (6) ??Heleď, právě teď Petr nachází svůj klíč. (= Malink’s (16a))
 look just now Petr find.IMPF his key
 ‘Look, Petr is finding his key right now.’

Choosing *finden* ‘find’ as a canonical example, Malink analyzes this verb in a two-dimensional logic as in (7), where the top formula of the two-dimensional array corresponds to the assertion and the bottom formula to the presupposition. Note that he abstracts away from the nominal arguments for simplicity.⁴

- (7)
$$\begin{bmatrix} \text{find}(e) \\ \neg \text{find}(e) \rightarrow \text{search}(e) \end{bmatrix} \quad (= \text{Malink's (23)})$$

He offers two options for the treatment of conative negation:

- (8)
$$\neg_{ass} \begin{bmatrix} A \\ B \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \neg A \\ B \end{bmatrix} \quad (= \text{Malink's (24)})$$

³ The form *nachází* ‘finds.IMPF’ may only receive an habitual or a historical present interpretation.

⁴ See Bergmann (1981) for background on the kind of two-dimensional logic adopted.

$$(9) \quad \neg_{ass2} \begin{bmatrix} A(e) \\ B \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \neg \exists e[A(e)] \\ B \end{bmatrix} \quad (= \text{Malink's (26)})$$

The operator \neg_{ass} in (8) applies to a two-dimensional array and simply negates the assertion, whereas the operator \neg_{ass2} in (9) requires as input an open formula of events e of type A and yields as output the assertion that there is no such event. Applied to the analysis of *finden* in (7), these two operators yield the following two arrays:

$$(10) \quad \begin{bmatrix} \neg \text{find}(e) \\ \neg \text{find}(e) \rightarrow \text{search}(e) \end{bmatrix} \quad (= \text{Malink's (25)})$$

$$(11) \quad \begin{bmatrix} \neg \exists e[\text{find}(e)] \\ \neg \text{find}(e) \rightarrow \text{search}(e) \end{bmatrix} \quad (= \text{Malink's (27)})$$

As Malink points out, both of these analyses entail that the event e (where free) is a searching event. Consequently, these two arrays are equivalent to the following two in which the presupposition that e is a searching event is expressed point-blank:

$$(12) \quad \begin{bmatrix} \neg \text{find}(e) \\ \text{search}(e) \end{bmatrix}$$

$$(13) \quad \begin{bmatrix} \neg \exists e[\text{find}(e)] \\ \text{search}(e) \end{bmatrix}$$

Since searching events are activities, the imperfective viewpoint aspect applies by default. In this way, the aspectual shift triggered by conative negation from the achievement meaning of *finden* to the activity meaning of *suchen* ‘search for’ is accounted for. As Malink puts it (following his (25)), ‘[...] the eventuality e which is projected by the VP into the AspP is a protracted searching activity.’

3. Comments on Malink

Malink’s proposal consists in a clever application of two-dimensional logic to the phenomenon of right boundary achievements under conative negation. While his proposal is original and thought-provoking, I will describe two potential problems that it faces. In doing so, I will be obliged to fill in certain details of his analysis that he himself does not fill in, which naturally leaves open the possibility that there is another way of working out his analysis that is immune to the critical points that I make in the next two sections.

3.1. Are presuppositions modifiable?

The more general problem is that Malink's analysis apparently makes crucial use of the idea that a presupposition can be modified or changed independently of the corresponding assertion. Although Malink does not present this as anything exceptional, it seems to me that it is. As far as I am aware, the following principle has a reasonable chance of being valid and should be entertained as a substantive universal of natural language unless a compelling empirical reason dictates otherwise:

- (14) Given an assertive meaning α , and its corresponding presuppositional meaning β , β may not be modified independently of α in the course of a semantic derivation, i.e., β may not be modified unless α is modified in the same way.⁵

In terms of two-dimensional logic, this amounts to a prohibition against an operator being able to manipulate the presupposition of a two-dimensional array independently of the assertion.⁶

To see that one natural way of spelling out Malink's proposal violates this principle, recall his analysis of the conatively negated form of *finden* 'find' in (13),⁷ which asserts that there is no finding event and presupposes that the event e is a searching event. It is clear from his discussion that the imperfective viewpoint operator should apply to this array, though he unfortunately does not explicitly show how this is to be done. Nevertheless, the following imperfective viewpoint aspect operator appears to be fit for the purpose that he has in mind:

$$(15) \quad \text{IMPF-2} \left[\begin{array}{c} A \\ B(e) \end{array} \right] \stackrel{\text{def}}{=} \left[\begin{array}{c} A \\ \exists e[B(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

This operator takes a two-dimensional array in which the presupposition is represented as an open formula of events e of type B and yields a two-dimensional array in which the presupposition is represented as an open formula of reference times t_r such that there is an event e of type B and t_r is a part of the run time $\tau(e)$ of e . Applied to the array in (13), IMPF-2 yields the following result:

⁵ Observe that we want to be able to modify presuppositions in tandem with the corresponding assertions. For example, suppose that x is *hungry* presupposes x *exists*, with the corresponding representation $\left[\begin{array}{c} \text{hungry}(x) \\ \text{exist}(x) \end{array} \right]$ in two-dimensional logic. If x is replaced with the constant *peter* to yield $\left[\begin{array}{c} \text{hungry}(\text{peter}) \\ \text{exist}(\text{peter}) \end{array} \right]$, then of course the presupposition has been modified, but this is permissible because the assertion has been modified in the same way.

⁶ Martin (2006, 312–313) also makes use of a restriction against the modification of presuppositions, though in a different context.

⁷ The same point could be made for the negated form of *finden* in (12), but the fact that e is free in the assertion here would complicate matters.

$$(16) \quad \left[\begin{array}{c} \neg \exists e[\text{find}(e)] \\ \exists e[\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

In this array, the assertion is that there is no finding event and the presupposition is represented as an open formula of reference times t_r such that there is a searching event e and t_r is a part of the run time $\tau(e)$ of e .

Clearly, IMPF-2 violates the principle in (14), because it modifies the presupposition independently of the assertion. As witnessed in the difference between the presuppositions of the arrays in (13) and (16), IMPF-2 changes the open formula of searching events e into the open formula of reference times t_r such that there is a searching event whose run time includes t_r .

In addition, I note that IMPF-2 would not work for the derivation of garden variety imperfective sentences in which the assertive component is imperfective, e.g.:

- (17) Peter sucht den Schlüssel.
 Peter searches-for the key
 ‘Peter is searching for the key.’

For these cases, Malink would need the following imperfective viewpoint operator (cf. (15)):

$$(18) \quad \text{IMPF} \left[\begin{array}{c} A(e) \\ B \end{array} \right] \stackrel{\text{def}}{=} \left[\begin{array}{c} \exists e[A(e) \wedge t_r \sqsubseteq \tau(e)] \\ B \end{array} \right]$$

Evidently, the only difference between IMPF-2 and IMPF is that the former modifies the presupposition, whereas the latter modifies the assertion; otherwise, their content is the same. Assuming that *suchen* ‘search for’ is represented as in (19), where ‘ \emptyset ’ indicates the lack of a presupposition,⁸ then the result of applying IMPF to this array is shown in (20).

$$(19) \quad \left[\begin{array}{c} \text{search}(e) \\ \emptyset \end{array} \right]$$

$$(20) \quad \left[\begin{array}{c} \exists e[\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \\ \emptyset \end{array} \right]$$

Naturally, it seems undesirable to have two imperfective viewpoint operators, IMPF-2 and IMPF, which differ only in which component (the presupposition or assertion) of a two-dimensional array they modify. If the principle in (14) were respected, IMPF-2 would be prohibited and IMPF would be the only permissible imperfective viewpoint operator with this content.

⁸ Whether or not *suchen* actually has a presupposition is irrelevant to the present point.

Malink (pers. comm.) has suggested that another option to consider would be that the array in (13) (or in (12), for that matter) is ‘leveled’ before the imperfective viewpoint operator applies. More precisely, such ‘leveling’ might consist in moving the presupposition to the assertion and conjoining it with the formula that previously formed the assertion. If so, the result of ‘leveling’ when applied to the array in (13) would be as follows:

$$(21) \quad \left[\begin{array}{c} \neg \exists e [\text{find}(e)] \wedge \text{search}(e) \\ \emptyset \end{array} \right]$$

Notice that the imperfective viewpoint operator IMPF would now suffice and there would be no need for the other imperfective viewpoint operator IMPF-2, which is an apparent advantage over the account just outlined.

However, this alternative formulation would have its price. It is not clear what independent motivation there would be for ‘leveling’, and unless it could be independently motivated, it would be a costly mechanism to postulate. Furthermore, I would argue that the principle in (14) would still be violated, because the removal of a presupposition would certainly count as a modification of the presupposition in the intended sense (unless, of course, the assertion were also removed!). Consequently, for the time being at least, I would find this option less attractive than the one discussed above.

3.2. How useful is this presupposition?

As Malink emphasizes, his strategy is to have a simple analysis of the conative negation operator and a relatively complex analysis of the verb phrase to which it applies. Indeed, ‘conative negation’ in his approach is really a misnomer, because there is in fact nothing ‘conative’ about his negation operator as defined in (8) or (9). He regards this as an advantage, but the ultimate cost of his approach lies in the presupposition he chooses for *finden*, and I want to argue that this presupposition not only lacks sufficient motivation but that it is even undesirable.

Let us consider how Malink would analyze positive sentences with *finden*. Although he does not work out such an example, it is reasonable to think that a perfective viewpoint operator is needed which existentially binds the event argument of the assertion and locates the run time of the event within the reference time (cf. (18)):

$$(22) \quad \text{PF} \left[\begin{array}{c} A(e) \\ B \end{array} \right] \stackrel{\text{def}}{=} \left[\begin{array}{c} \exists e [A(e) \wedge \tau(e) \sqsubseteq t_r] \\ B \end{array} \right]$$

Applying PF to the array in (7), we obtain the following result:

$$(23) \quad \left[\begin{array}{c} \exists e [\text{find}(e) \wedge \tau(e) \sqsubseteq t_r] \\ \neg \text{find}(e) \rightarrow \text{search}(e) \end{array} \right]$$

However, notice that the free variable e in the presupposition is now no longer free in the assertion, which has the effect of completely disassociating the presupposition from the assertion. As a statement in its own right, the presupposition is true for some values of e but false for most others (for an instance of the latter, just choose any event that is neither a finding event nor a searching event). Yet this means that the truth value of the assertion is effectively independent of the truth value of the presupposition: in particular, the assertion may be true (that there is a finding event whose run time is included in the given reference time) even if the presupposition is false (for a value of e). Technically, this kind of situation (i.e., a true assertion with a false presupposition) can arise in two-dimensional logic, but generally it is admissible only in the case of complex formulas with binary connectives (see Bergmann 1981), which is not the case here.⁹ Clearly, then, something seems to have gone awry.¹⁰

The problem is that Malink's presupposition for *finden* plays a role only in conative negation; otherwise, it is not motivated and even gets in the way, as we have just seen. But then it is illusory to think that something is gained by attributing such a presupposition to *finden*. Instead, it would be more appropriate to try to account for the aspectual shift witnessed with conative negation more directly.

4. An alternative proposal

Is there a way to account for the phenomenon of conative negation without running into the two problems just described? In this section, I will sketch such an account, highlighting the ways in which it essentially differs from Malink's analysis.

To begin with, let us assume that *finden* 'find' has a sense in which it presupposes a searching event (recall (1) and (2)).¹¹ The first task is to produce a representation for *finden* that captures this:

$$(24) \quad \left[\begin{array}{l} \text{find}(e) \wedge \tau(e) \sqsubseteq t_r \\ \exists e[\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

⁹ More precisely, if the aim is to preserve the *intended* connection between an assertion and its corresponding presupposition in the case of simple formulas (namely, that the failure of a presupposition signifies semantic anomaly), then the *admissible valuations* for simple formulas should not allow for an assertion to be true if its corresponding presupposition is false. As Bergmann (1981) emphasizes, a choice of admissible valuations is dictated by a choice of *presuppositional policies*, and the latter choice may of course be questioned. My criticism of the result in (23) is that it breaks the intended connection between the assertion and its corresponding presupposition, which violates the presuppositional policy (for simple formulas) that the failure of a presupposition should signify semantic anomaly.

¹⁰ Although it is true that the account with 'leveling' mentioned in the previous section (see (21)) would not face this problem, it would pay a compensatory price with the postulation of the costly mechanism of 'leveling'.

¹¹ This seems to be in accordance with Malink's description of his intuitions in section 1 of his paper.

In this array, the assertion is that e is a finding event whose run time $\tau(e)$ is included in the reference time t_r and the presupposition is that there is a searching event e whose run time $\tau(e)$ includes t_r . The need for the reference time will become more evident below when we consider negation. Note that searching events may be defined as trying-to-find events much in the spirit of Montague (1974):¹²

$$(25) \quad \text{search}(e) \wedge t_r \sqsubseteq \tau(e) \stackrel{\text{def}}{=} \text{try}(e, \lambda e' [\text{find}(e') \wedge \tau(e') \sqsubseteq t_r]) \wedge t_r \sqsubseteq \tau(e)$$

Finding events are the right boundaries of searching events. This is ensured by the following axiom, where ‘ $\text{r-b}(e, e')$ ’ indicates that e is the right boundary of e' :¹³

$$(26) \quad \forall e \forall t_r [(\text{find}(e) \wedge \tau(e) \sqsubseteq t_r) \rightarrow \exists e' [\text{search}(e') \wedge t_r \sqsubseteq \tau(e') \wedge \text{r-b}(e, e')]]$$

This axiom evidently renders the presupposition in (24) redundant in the case of a positive sentence. However, the presupposition is not redundant in the case of a negative sentence, as we will soon see.

The next step is to show how positive sentences with *finden* are derived (ignoring tense, for simplicity). This consists in existentially binding the event variable of the assertion:

$$(27) \quad \mathcal{E} \left[\begin{array}{c} A(e) \\ B \end{array} \right] \stackrel{\text{def}}{=} \left[\begin{array}{c} \exists e [A(e)] \\ B \end{array} \right]$$

Applied to the array in (24), the event binding operator \mathcal{E} yields the following result:

$$(28) \quad \left[\begin{array}{c} \exists e [\text{find}(e) \wedge \tau(e) \sqsubseteq t_r] \\ \exists e [\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

In this array, the assertion states that there is a finding event whose run time is included in the reference time t_r , whereas the presupposition specifies that there is a searching event whose run time includes t_r . In the limiting case where the (instantaneous) run time of the finding event is identical with t_r , t_r is simply the right boundary of the run time of the searching event.

¹² Though note that Montague defines *seek* instead of *search for*. Strictly speaking, the relation *try* in (25) should probably apply to the *intension* of $\lambda e' [\text{find}(e') \wedge \tau(e') \sqsubseteq t_r]$, but for simplicity I keep things extensional here.

¹³ A more accurate rendition of this axiom would make use of the relation *end* from Piñón (1997), i.e.,

$$\forall e [(\text{find}(e) \wedge \tau(e) \sqsubseteq t_r) \rightarrow \exists e' [\text{end}(e, e', \lambda e'' [\text{search}(e'') \wedge t_r \sqsubseteq \tau(e'')])]],$$

where ‘ $\text{end}(e, e', \lambda e'' [\text{search}(e'') \wedge t_r \sqsubseteq \tau(e'')])$ ’ means that e is the end of a searching event e' whose run time $\tau(e')$ includes the reference time t_r . In that framework, ends are always right boundaries but not vice versa.

Ordinary negative sentences are formed with the help of the following negation operator:

$$(29) \quad \neg \left[\begin{array}{c} A \\ B \end{array} \right] \stackrel{\text{def}}{=} \left[\begin{array}{c} \neg A \\ B \end{array} \right]$$

Observe that the negation operator \neg has the same content as Malink's conative negation operator \neg_{ass} in (8). Furthermore, the combination of \mathcal{E} followed by \neg has the same effect as Malink's second conative negation operator \neg_{ass2} in (9). However, none of this is surprising given that both \neg and \neg_{ass} are just two designations for the standard negation operator.

Applied to the array in (28), the negation operator \neg delivers the following result:

$$(30) \quad \left[\begin{array}{c} \neg \exists e [\text{find}(e) \wedge \tau(e) \sqsubseteq t_r] \\ \exists e [\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

Here, the need for the presupposition becomes evident because the axiom in (26) does not ensure the existence of a searching event in the absence of a finding event. Moreover, the run time of the presupposed searching event is constrained to include the reference time t_r , which has the consequence that a searching event (with the same agent, etc.) earlier or later than the reference time would not satisfy the presupposition.

Thus far, I have presented an analysis of *finden* that includes a presupposition of the existence of a searching event and have shown how positive and negative sentences are derived from it. However, this is still short of treating the phenomenon of conative negation that Malink discusses. For the latter, I need to introduce a special conative negation operator, defined as follows:

$$(31) \quad \neg_{con} \left[\begin{array}{c} A(t_r, e) \\ B \end{array} \right] \stackrel{\text{def}}{=} \left[\begin{array}{c} \neg \exists e [A(t_r, e)] \wedge \text{try}(e', \lambda e'' [A(t_r, e'')]) \wedge t_r \sqsubseteq \tau(e') \\ B \end{array} \right]$$

The conative negation operator \neg_{con} modifies the assertion by negating the existence of events of type A at the reference time t_r and by introducing trying-to- A events e' whose run times $\tau(e')$ include t_r . For an illustration of \neg_{con} in action, consider the result that it yields when applied to the array in (24):

$$(32) \quad \left[\begin{array}{c} \neg \exists e [\text{find}(e) \wedge \tau(e) \sqsubseteq t_r] \wedge \text{try}(e', \lambda e'' [\text{find}(e'') \wedge \tau(e'') \sqsubseteq t_r]) \wedge t_r \sqsubseteq \tau(e') \\ \exists e [\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

This array represents the conatively negated form of *finden*, which appears in (3a). The assertive component states both that there is no finding event whose run time is included in the reference time t_r and that the event e' is a trying-to-find event whose run time includes t_r . The presuppositional component is the

same as before, requiring that there be a searching event e whose run time $\tau(e)$ includes t_r .

Observe that the array in (32) is equivalent to the following one by virtue of the definition in (25):

$$(33) \quad \left[\begin{array}{l} \neg \exists e [\text{find}(e) \wedge \tau(e) \sqsubseteq t_r] \wedge \text{search}(e') \wedge t_r \sqsubseteq \tau(e') \\ \exists e [\text{search}(e) \wedge t_r \sqsubseteq \tau(e)] \end{array} \right]$$

Here, the assertion is both that there is not a finding event whose run time is included in the reference time t_r , and that the event e' is a searching event whose run time $\tau(e')$ includes t_r . This shows how the conatively negated form of *finden* can entail the meaning of *suchen* ‘search for’ in the assertion without an appeal to the presupposition. Note that, although the presupposed searching event and the searching event e' of the assertion temporally overlap (given that t_r is a part of both of their run times), the interpretation of this array alone does not force them to be identical.¹⁴

I want to emphasize that the viability of the present approach does not depend on the definition in (25). If it turned out that searching events are not just trying-to-find events but rather more specific in nature, then the array in (32), which represents the conatively negated form of *finden*, would not be equivalent to the one in (33). But this would simply mean that the conatively negated form of *finden* designates trying-to-find events that may be less specific in nature than searching events. More concretely, it would be feasible to replace the definition in (25) with an axiom saying that every searching event is a trying-to-find event (but not vice versa), in which case the array in (33) would imply the one in (32), for values of e' and t_r (but not vice versa).

The present account succeeds in avoiding the two potential problems for Malink’s approach discussed in the previous section. Firstly, it does not make use of a mechanism that modifies a presupposition independently of the corresponding assertion, thereby respecting the principle in (14). In particular, the presupposition of *finden* of the existence of a searching event, as represented in (24), is not touched by any operator. In contrast, the two ways of working out Malink’s approach discussed in section 3.1 do violate this principle, although of course there may still be another formulation which does not. Secondly, the presupposition of the existence of a searching event plays an intuitive role (and does not cause concern) in garden variety positive and negative sentences (see (28) and (30)), as well as cases of conative negation (see (32)), even if it does not figure centrally in the analysis of the latter. In contrast, as argued in section 3.2, one natural way of spelling out Malink’s proposal renders his presupposition for *finden* worrisome

¹⁴ However, if there were an additional principle which required any two temporally overlapping searching events e_1 and e_2 with the same agent, etc. to be parts (though not necessarily proper parts) of a searching event e_3 , with a limiting case in which e_1 , e_2 , and e_3 are all identical, then the two searching events in (33) (e' of the assertion and e of the presupposition) minimally would both be parts of a bigger searching event and in the limiting case would be identical.

for anything but conative negation. Finally, the present account, in contrast to Malink's, attributes real conative content to conative negation.

In closing, it may be best to regard the conatively negated form of *finden* in (32) in lexicalist terms as a kind of 'negative verb'. This idea is supported by the Czech data, where—as Malink points out in connection with (5a) and (6)—the verb form *nenachází* 'NEG.finds.IMPF' cannot be viewed as the negated version of *nachází* 'finds.IMPF' semantically. Instead, what the conative negation operator negates here is closer to the meaning of *najde* 'finds.PF' (cf. (5b)), but with a result that includes an imperfective meaning (a trying-to-find) as opposed to merely a negated perfective meaning (a non-finding), pretty much as displayed in (32).

Another reason to regard the conative negation operator \neg_{con} as lexically restricted is that it is confined to right boundary achievements. For example, Malink observes that the following accomplishment sentences do not allow for a conatively negated reading:

- (34) a. Peter schreibt den Brief nicht. (= Malink's (9a))
 Peter writes the letter NEG
 'Peter doesn't write the letter.'
 b. Peter repariert des Fahrrad nicht. (= Malink's (9b))
 Peter repairs the bicycle NEG
 'Peter doesn't repair the bicycle.'

Just as (34a) cannot be understood to mean that Peter is trying to write the letter, (34b) cannot mean that he is trying to repair the bicycle, which indicates that \neg_{con} does not apply to *schreiben* 'write' or *reparieren* 'repair'. A straightforward way of treating this would be to attribute to \neg_{con} a 'selectional restriction' that is specified for right boundary achievements, where a 'right boundary achievement' could be defined as a sort of event predicate.

In any case, insofar as this is considered a problem for the present account, a similar problem arises for Malink's approach, because he would have to ensure that verbs other than right boundary achievements lack a presupposition of the kind that would give rise to an aspectual shift under conative negation. To drive this point home, in his approach there is no independently evident reason why *suchen* should not be analyzed as $\left[\begin{array}{c} \text{search}(e) \\ \neg\text{search}(e) \rightarrow \text{find}(e) \end{array} \right]$ (cf. (7)), and yet this would yield the meaning of *finden* under conative negation, which is undesirable. Therefore, Malink would have to regulate the phenomenon of conative negation in his account via a careful distribution of the kind of presupposition that plays a crucial role in conative negation.

References

- Bergmann, Merrie. (1981): Presupposition and two-dimensional logic. *Journal of Philosophical Logic* 10, 27–53.
- Heyde-Zybatow, Tatjana. (2004): Untersuchungen zur Verbklassse der Achievements. *Grammatik und Kontext: Zur Interaktion von Syntax, Semantik und Prosodie bei der Informationsstrukturierung* (= Steube, A. (ed.): *Linguistische Arbeitsberichte* 81), Institut für Linguistik, Universität Leipzig, 11-32
- Malink, Marko. (2007): Right boundary achievements under conative negation. This volume.
- Martin, Fabienne. (2006): *Prédicats statifs, causatifs et résultatifs en discours: Sémantique des adjectifs évaluatifs et des verbes psychologiques*. Doctoral dissertation, Université libre de Bruxelles.
- Montague, Richard. (1974): On the nature of certain philosophical entities. In: Thomason, R.H. (ed.), *Formal philosophy: Selected papers of Richard Montague*. New Haven: Yale University Press, 148–187.
- Piñón, Christopher. (1997): Achievements in an event semantics. In Lawson, A. and Cho, E. (eds.): *Proceedings of SALT VII*. Ithaca: CLC Publications, 273–296.
Also available at <http://pinon.sdf-eu.org/covers/aes.html>.