

The semantic contribution of past tense morphology in Palestinian

A central question in the study of counterfactuality concerns the semantic contribution of past tense morphology. As is well known, past tense morphemes can be used to express both semantic past tense and counterfactuality. Basing ourselves on data from Palestinian, which is more transparent than other languages when it comes to expressing counterfactuality, we demonstrate that past tense morphemes unambiguously mark that the truth of a particular proposition is not guaranteed at the time of utterance. Take the following data:

- (1) kaanat farhaane.
 be.PST.3.SG.F happy.ADJ.SG.F ‘She was happy.’
- (2) a. Kaanat bitkuun farhaane.
 be.PST.3.SG.F MOD-be.IMPFV.1.SG.F happy.ADJ.SG.F ‘She would be happy.’
 b. kaanat b-tifrah
 be.PST.3.SG.F MOD-happy.IMPFV.1.SG.F ‘She would be happy’
- (3) kaanat firhat
 be.PST.3.SG.F happy.PFV.1.SG.F ‘She would have been happy’

The question immediately arises as to what the semantic properties of *kaan* are, such that both its past tense reading and its counterfactual reading follow compositionally. Previous approaches differ w.r.t. the semantic status of elements comparable to *kaan*. Ippolito (2003/6) takes past tense markers to indicate ‘real past’, which constrains epistemic accessibility relations, but see Schulz (2007) for a number of problems that this account faces. On the other hand, Iatridou (2000) takes past tense morphemes to express exclusion either from the present time or the actual world (hence indicating either past tense or counterfactuality). For Iatridou, the past morpheme itself is semantically active. However, recent observations indicate that past tense morphology is semantically vacuous and only functions as markers for abstract operators (cf. von Stechow 2006, Zeijlstra 2009). In this paper, we argue that past tense morphemes are neither ‘real past’, nor ambiguous exclusive features. Instead, we argue that all instances of past tense morphology are overt markers of the presence of an abstract operator entailing Non-Actual Veridicality (NAV), where NAV is defined as in (4).

$$(4) \quad ||NAV|| p(w^\circ)(t^\circ) = 1 \text{ iff } \exists w' . \exists t' [w' \neq w^\circ \ \& \ t' \leq t^\circ \ \& \ [p(w^\circ)(t') \vee p(w')(t^\circ)]]$$

The grammatical contribution of *kaan* is that it must agree with an operator at least as strong as NAV. Implemented in canonical agreement systems (Chomsky 1995, Pestersky & Torrego, 2007, von Stechow 2006, Zeijlstra 2009), this entails that *kaan* carries an uninterpretable feature [uNAV], which has to be checked against an element carrying interpretable [iNAV]. Two candidates for such operators suggest themselves. The first candidate is an NAV mood operator (Op_{NAV}) carrying [iNAV] with the semantics in (4) that is therefore able to check *kaan*’s [uNAV] feature. But also a true past tense operator (Op_{PAST}), carrying [iPAST] is able to check *kaan*’s [uNAV]. Past tense entails NAV and, as Bejar (2009) has shown, if some feature entails another feature then it is also able to check off the uninterpretable counterpart of that feature. This explains why *kaan* can have both a past tense and a counterfactual interpretation. However, a number of questions arise as well: (i) why is the default interpretation of *kaan* past tense; (ii) why does a sentence like (2) suggest that the proposition is false in the actual world (note that this does not straightforwardly follow from NAV); and (iii) if *kaan* can have its feature checked by two different operators in the syntax (the mood operator generally taken to be hosted in MoodP and the tense operator in TP), is that distinction also manifested in the morpho-syntax itself?

Concerning (i), the answer lies in the syntax where MoodP always selects TP. Consequently, it is impossible that Op_{NAV} can be included in a sentence without a particular tense operator such as Op_{PAST} in TP. If no other tense marker is present than *kaan*, *kaan* must agree with the Op_{PAST} , and the counterfactual reading is absent. This predicts that if another tense operator is present then the *kaan* may check its [uNAV] either against Op_{PAST} or against Op_{NAV} , yielding

