AMG: Case Theory in Montague Grammar

Kiyong Lee
Korea University
Seoul 132

0. The present paper is part of my continued efforts to establish a theory of case in Montague grammar and to account for case assignment and word order in case-marked languages. In my earlier papers(1981a,b; 1982a, b,c), I proposed RPTQ, a revised version of Montague's PTQ, to treat case marking and reordering in verb-final languages like Korean. But in RPTQ case marking was restricted to noun phrases, and so was reordering. In the present paper, I will propose a more generalized version AMG, an augmented Montague grammar, to extend case marking and reordering to other types of phrases, in particular verbal or sentential complements. I will then show how AMG works for a small fragment of Korean that includes negative constructions.

1. AMG is basically the same as RPTQ. Like RPTQ, it employs three basic devices of case assignment: (i) case indexing, (ii) case marking, and (iii) case shifting.

First, case indexing subcategorizes various types of verb phrases with respect to case indices. The IV-phrase onta-come, for instance, belongs to the category (1,0) denoting a set of individuals, where the nominative case index 1 corresponds to the semantic type *e* and the zero 0, to the semantic type *t*.

Secondly, case marking specifies the case-form of a *T*-phrase of the category $((n,0),0)_n$, or simply +n, by freely replacing the variable n with some case index 1 (Nom), 2(Acc) or 3(Dat) and then assigning to the *T*-phrase a case particle corresponding to the case index. For example, Mary_{+n} may change to Mary-ka_{+1}, where the number 1 is the nominative case index and ka, a nominative case particle.

Thirdly, case shifting derives complex verbs and reassigns them a new category by reshuffling case indices. The complex causative verb čuki-kill is derived from čuk-die of the category (1,0) and is reassigned a new category (+2,(1,0)).

2. AMG differs from RPTQ in further generalizing the set of categories. The set of categories in AMG is defined as follows:

(1) Categories

a. 0, n $\in$ Cat , where n = 1, 2, 3, . . .

b. A $\in$ Cat $\rightarrow$ A', A_n $\in$ Cat

c. A, B $\in$ Cat $\longrightarrow$ (A,B) $\in$ Cat

These categories correspond to the following semantic types:

(2) Category-Type Correspondence[1]

a. $f(0) = t$, $f(n) = e$

b. $f(A) = f(A') = f(A_n)$

c. $f(A,B) = \langle\langle s, f(A)\rangle, f(B)\rangle$

The zero 0 corresponds to t denoting a truth-value and each natural number n to the type e denoting an entity. Neither the prime ' nor the subscript n has any semantic role. Finally, the category (A,B) corresponds to the type denoting a function from objects of the semantic type f(A) to objects of the semantic type f(B).

Here are some common categories and expressions of Korean that belong to them:

(3) Common Categories

Categories	Mnemonic	Description	Expressions
0		sentence	
(1,0)	IV	action verb	o-come
(1,0)'	DIV	description verb	palk-bright
$((n,0),0)_n$	T, or +n	noun phrase	Mary
(+2,(1,0))	TV	transitive verb	salanha-love

By the convention of right associativity, we erase any parentheses or commas that create no amibiguity: e.g., +210 for (+2,(1,0)), 10 for (1,0), but (10)' for (1,0)'.

3. Both RPTQ and AMG are versions of categorial grammar that combines two well-formed phrases into one provided that one is of the category A and the other, of the category B/A. By relaxing this provision, RPTQ and AMG concatenate two well-formed phrases only if one phrase is of the category A and the other, of the category (... A ...). Hence the following concatenations are well-formed in RPTQ or AMG:

(4) [[Mary-ka]$_{+1}$ [ča-n-ta]$_{10}$]$_{0}$ 'Mary sleeps'

(5) [[Mary-ka]$_{+1}$ [[John-ɨl]$_{+2}$ [salaŋhanta]$_{+2\,10}$]$_{10}$]$_{0}$

'Mary loves John'

(6) [[John-ɨl]$_{+2}$ [[Mary-ka]$_{+1}$ [salaŋhanta]$_{+2\,10}$]$_{20}$]$_{0}$

But the following concatenation is ill-formed:

(7) [[John-ɨl]$_{+2}$ [čanta]$_{10}$]

Note in particular that sentences (5) and (6) are derived by simple concatenation without involving any scrambling or reordering transformation. One of the strongest motivations for RPTQ or AMG is to derive so-called scrambled sentences by categorial combination.

4. AMG extends case marking to sentential or verbal complements. Here are some examples:

(8) Mary-nɨn [[[pi-ka onta]$_{0}$-nɨn kəs]$_{0'_n}$-ɨl]$_{0'_2}$[molɨnta]$_{0'_2 10}$]$_{10}$

Top rain comes Comp Acc not know

'Mary doesn't know that it rains'

(9)[[Mary-ka]$_{+1}$ [[piano-lɨl čhi-l čul]$_{(10)'_n}$-ɨl]$_{(10)'_2}$ [anta]$_{(10)'_2 10}$]

play know

'Mary knows how to play the piano'

(10) latinə-nɨn [[uli-ka paeu-ki]$_{(20)'_n}$-ka]$_{(20)'_1}$[əlyəpta]$_{(20)'_1 10}$]]

we learn difficult

'Latin is difficult for us to learn'

In deriving theses sentences, AMG derives sentential or verbal complements that potentially recieve case markers. Just like the term phrase category $+_n$, potential case receivers are marked with the subscript n.

Some complements are never assigned a case.

(11)[Mary-nɨn [[[pi-ka onta]$_{0}$-ko]$_{0'}$ [malhaessta]$_{0' 10}$]$_{10}$]$_{0}$

Comp said

'Mary said that it rained'

Thus the following strings are ill-formed:

(12) * Mary-ka [[pi-ka onta-ko]$_{0'}$ [molɨnta]$_{0'_2 10}$]

(13) * Mary-ka [[[pi-ka onta-ko]$_{0'}$-lɨl]$_{0'_2}$ molɨnta]]

(12) violates the categorial condition on concatenation and (13) the subscript condition on case marking.

Sentential complements may be reordered:

(14) [[pika ontanɨn kəs-ɨl]$_{0'2}$ [[Mary-ka]$_{+1}$ [molɨnta]]$_{0'210}$]$_{0'20}$]

(15) [[pika onta-ko]$_{0'}$ [[Mary-ka]$_{+1}$ [malhaessta]$_{0'10}$]$_{0'0}$]$_{0}$

These sentences are again obtained by ordinary concatenation.

However, verbal complements must not be reordered:

(16) *[[piano-lɨl čhi-l čul-ɨl]$_{(10)'2}$ Mary-ka anta]

(17) *[[uli-ka paeuki-ka]$_{(20)'1}$ latinə-ka əlyəpta]

These two are not acceptable, although they do not violate the categorial condition on concatenation. (This may be explained by introducing the notion of control into Motague grammar.)

5. Negative constructions in Korean present some interesting problems for Montague grammar. I will show that AMG deals with these problems in a rather simple and elegant manner.

There are three types of negative constructions in Korean:

(18) a. pi-ka nun-ita
rain Nom snow be
'Rain is snow'

b. pi-ka nun-i anita (Type 0)
snow Nom not be
'Rain is not snow'

(19) a. pi-ka onta

come

'Rain comes' = 'it rains'

b. pi-ka ani onta (Type 1)

not

'It doesn't rain'

c. pi-ka o-či anihanta (Type 2)

come Comp not do

'It doesn't rain'

Negative sentence (18b) corresponds to affirmative sentence (18a); negative sentences (19b) and (19c) both having the same meaning correspond to affirmative sentence (19a).

5.1 Type 0 negative sentence (18b) may be analyzed as in:

(20) $[[\text{pi-ka}]_{+1}\ [[\text{nun-i}]_{+1}\ [\text{anita}]_{+110}]_{10}\ ]_{0}$

k1

k+1

This analysis is obtainable in AMG by applying the operations of concatenation, k1 and k+1 to the negative copular verb anita of the category +110. This verb then denotes a two-place non-identity relation between an individual and a property set, having the following translation:

(21) $\text{anita}_{+110} \Rightarrow \lambda\mathcal{P}\, \lambda z\, \neg\mathcal{P}\{\hat{y}[z = y]\}$

On the basis of (21), analysis (22) translates into:

(22) $\lambda PP\{r\}\ ({}^{\wedge}\lambda\mathcal{P}\lambda z \neg \mathcal{P}\{\hat{y}\ [z = y]\}\ (\hat{P}P\{s\}))$

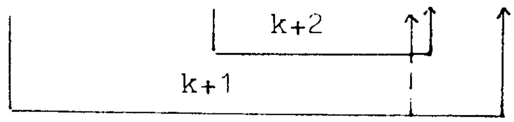
<u>or</u> $\neg[r = s]$, where pi-<u>rain</u> $\Rightarrow \lambda PP\{r\}$

nun-<u>snow</u> $\Rightarrow \lambda PP\{s\}$

This translation correctly represents the logical form of negative sentence (18b).

In contrast to the well-formed analysis (20), the following analysis is unacceptable:

(23)* $[[\text{nun-i}]_{+1}\ [[\text{pi-ka}]_{+1}\ [\text{anita}]_{+110}\ ]_{10}\ ]_{0}$



Concatenations $^{k+2}$ and $^{k+1}$ do not violate the categorial condition on concatenation in AMG.[1] Concatenation k+1 comes to violate the principle of variable collision in the process of translating (23) into logic, for both of the term phrases are marked with the same category index. This principle thus blocks reordering in double-nominative sentences like (23).

Type 0 negation creates scope amibiguity, when the negative copular verb takes a quantified phrase as its subject argument. Consider:

(24) motɨn salam-i papo-ka anita

all man fool not be

'All men are not fools'

This sentence has two readings, a wide scope and a narrow scope reading:

(25) motɨn salam-i papo-ka anila John-man-i papo-ta
only

'Not every man is a fool, but only John is' (WR)

(26) motɨn salam-i papo-ka aniəsə motu hapkyə khaessta
all passed

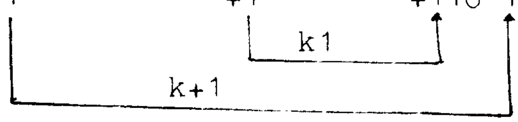
'No one was a fool, so all passed' (NR)

This amibiguity of type 0 negation can easily be accounted for in AMG. Two types of anita are set up: one is of the category +110 and another is of the category +1+10. While the former denotes a two-place relation between an individual and a property set, the latter denotes a two-place relation of non-identity between two property sets, having the following translation:

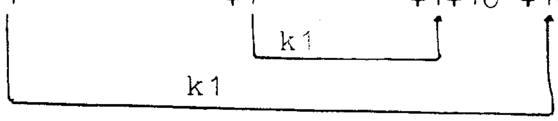
(27) $anita_{+1+10} \Longrightarrow \lambda\mathcal{P}_1 \lambda\mathcal{P}_2 \neg\mathcal{P}_1\{\hat{y}\,[\mathcal{P}_2\{\hat{z}\,[\,z = y\,]\}\,]\}$

Because of the lexical ambiguity of anita[2], sentence (24) may have two distinct analyses:

(28) [[motɨn salam-i]$_{+1}$ [[papo-ka]$_{+1}$ [anita]$_{+110}$]$_{10}$]$_{0}$



(29) [[motɨn salam-i]$_{+1}$ [[papo-ka]$_{+1}$ [anita]$_{+1+10}$]$_{+10}$]$_{0}$



These concatenations are again well-defined, observing the categorial condition on concatenation.

These analyses then give the following non-equivalent translations:

(30) $\lambda P \bigwedge x[\text{salam}'(x) \rightarrow P\{x\}](^\wedge\text{anita}'\ (^\wedge\lambda P \bigvee y[\text{papo}'(y) \wedge P\{y\}]))$

<u>or</u> $\bigwedge x[\text{salam}'(x) \rightarrow \neg \text{papo}'(x)]$

(31) $[\text{anita}'(^\wedge\lambda P \bigvee y[\text{papo}'(y) \wedge P\{y\}])](^\wedge\lambda P \bigwedge x[\text{salam}'(x) \rightarrow P\{x\}])$

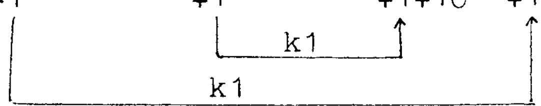
<u>or</u> $\neg \bigvee x[\text{papo}'(x) \wedge \bigwedge y[\text{salam}'(y) \rightarrow x = y]]$

<u>or</u> $\bigwedge x[\text{papo}'(x) \rightarrow \bigvee y[\text{salam}'(y) \wedge \neg x = y]]$

Narrow scope reading (30) is true when there is no man who is a fool, while *wide* scope reading (31) is true when there is at least one man who is not a fool.

But the lexical amibiguity of *anita* does not create any amibiguity in a sentences like (18b). The following analysis of (18b), for instance, gives a translation equivalent to that of analysis of (20):

(32) $[[\text{pi-ka}]_{+1}\ [[\text{nun-i}]_{+1}\ [\text{anita}]_{+1+10}\ ]_{+10}\ ]_C$



(33) $\text{anita}'(\hat{P}P\{s\})(\hat{P}P\{r\})$, where *anita* $\Rightarrow$ (27)

It will easily be shown that (33) simplifies to (22) by lambda conversion. Despite the lexical amibiguity of *anita*, sentence (18b) is not ambiguous because it contains no logical operator or quantifier other than the negative verb *anita*.

5.2 Type 1 negative sentence (19b) may be analyzed in AMG as in the following:

(34) a. $[[\text{pi-ka}]_{+1}\ [\text{ani}\ [\text{onta}]_{10}\ ]_{10}\ ]_0$

k+1

b. $[[\text{pi-ka}]_{+1}\ [\text{ani}\ [\text{onta}]_{10}]_{+10}\ ]_0$

k1

These two analyses give translations (35a) and (35b), respectively, both of which are equivalent to (35c):

(35) a. $\hat{P}P\{r\}(\hat{}\,\text{ani}'(\hat{}\,\text{onta}'))$

b. $\text{ani}'(\hat{}\,\text{onta}')(\hat{P}P\{r\})$

c. $\neg\text{onta}'(r)$

These formulas are all true just in case that it does not rain.

In the analyses of (34a,b), I have set up two distinct categories of ani onta, 10 and +10, just as I assigned the copular verb anita two distinct categories +110 and +1+10. This distinction gives two different translations of ani onta:

(36) a. $\text{ani onta}_{10} \Rightarrow \lambda x \neg\, \text{onta}'(x)$

b. $\text{ani onta}_{+10} \Rightarrow \lambda \mathcal{P} \neg \mathcal{P}(\hat{}\,\text{onta}')$

Furthermore, this distinction accounts for the scope amibiguity of type 1 negative sentences like:

(37) motɨn namča-ka ani onta

all male come

'All men do not come'

This sentence will have the following two distinct analyses:

(38) a. $[[\text{mot}\dot{\text{i}}\text{n namča-ka}]_{+1}\ [\text{ani}\ [\text{onta}]_{10}\ ]_{10}\ ]_{0}$

k+1

b. $[[\text{mot}\dot{\text{i}}\text{n namča-ka}]_{+1}\ [\text{ani}\ [\text{onta}]_{10}\ ]_{+10}\ ]_{0}$

k1

(38a) gives a narrow scope reading, meaning that no one comes.

(39) $\lambda P \bigwedge x[\text{salam}'(x) \rightarrow P\{x\}\ ](\hat{y} \neg \text{onta}'(y))$

or $\bigwedge x[\text{salam}'(x) \rightarrow \neg\, \text{onta}'(x)]$

On the other hand, (38b) gives a wide scope reading, meaning that not all men come.

(40) $\lambda \wp \neg \wp\ ({}^{\wedge}\text{onta}')(\hat{P} \bigwedge x[\text{salam}'(x) \rightarrow P\{x\}\ ]$

or $\neg \bigwedge x[\text{salam}'(x) \rightarrow \text{onta}'(x)]$

The negative particle ani may be treated as an adverb belonging to either the category (10)10 or the category (10)+10, thus allowing the following analyses:

(41) a. $[[\text{ani}]_{(\underline{10})10}\ [\text{onta}]_{\underline{10}}\ ]_{10}$

k'1

b. $[[\text{ani}]_{(\underline{10})+10}\ [\text{onta}]_{\underline{10}}\ ]_{+10}$

k'1

These analyses again observe the categorial conditon on concatenation.

But such a treatment also allows the following unacceptable derivation:

(42) $[[ani]_{(10)10}\ [pi\text{-}ka]_{+1}\ ]_{(\underline{10})0}\ [onta]_{\underline{10}}\ ]_0$

k'+2   k'1

Here, concatenations k'1 and K'+2 do not violate the categorial condition. Rather they need be allowed to account for the preposing of adverbs like manhi-much or č^{h}ənč^{h}ənhi-slowly.

(43) a. pi-ka manhi onta   'It rains much'

b. manhi pi-ka onta

(44) a. pi-ka č^{h}ənč^{h}ənhi onta

rain slowly come

'It rains slowly'

b. č^{h}ənč^{h}ənhi pi-ka onta

(43b) and (44b) are both acceptable where the adverbs manhi and č^{h}ənč^{h}ənhi are preposed. To derive (43b), for instance, we need both of the concatenations, k'1 and k'+2.

(45)[3] $[[[manhi]_{(10)10}\ [pi\text{-}ka]_{+1}\ ]_{(\underline{10})0}\ [onta]_{\underline{10}}\ ]_0$

k'+2   k'1

Translation: λP manhi'(P)(r)(^onta')

or [manhi'(^onta')](r)

Unlike these adverbs, the negative particle *ani* may not be separated from its verb. Here is another example:

(46) a. John-i Mary-lɨl ani salaŋhanta

'John does not love Mary'

b. * John-i[[ani]$_{(10)10}$ [Mary-lɨl salaŋnanta]$_{10}$]$_{10}$

k'1

(46b) is unacceptable, although concatenation k'l does not violate the categorial condition.

In order not to separate an adverb like *ani* from its verb, I employ the notion of basic, or lexical, category and allow the concatenation of such an adverb only with a basic verb. Note, however, that every lexical verb is a basic verb, but that basic verbs may include morphologically derived complex verbs like causatives or desideratives. Thus the following are acceptable:

(47) a. ani čuk -i - ta

not die Cause

'not die'

b. ani čap - hi - ta

not catch Passive

'be not caught'

c. ani ka-ko sip^{h}ta

go like

'not like to go'

On the other hand, the following are not acceptable:

(48) a. * čuk ani ita

b. * čap ani hi ta

c. * ka-ko ani sip^{h}ta

These strings are ill-formed because they violate the categorial condition on concatenation, whereas (47a,b, c) are acceptable because these complex verbs are treated as basic, not as phrasal.

The following rule schema derives type 1 negative verb phrases:

(49) Type 1 Negation

$$F_{k'i}([\alpha]_{(X10)X'10}, [\beta]_{X10}) = [\alpha\beta]_{X'10}$$

where i = 1, 2, 3, . . .

X, X' are strings(possibly null) of categories such that

X' = X or X+ ,

α = ani , and β is basic.

Translation: $\alpha'(\hat{}\beta')$

These rules are ordinary rules of functional application that observe the categorial condition on concatenation. But to make these rules operative, we must set up various types of ani in the lexicon. Each of its syntactic categories is of the form (X10)X'10 so that the negative particle ani is treated as an adverb, for the string of categories (X10) is a well-formed verbal category. The string X'10 in the strings of the categoy indices for ani makes ani an opaque verb, for X'10 is understood to be as either X10 or X+10. Because of this subcategorization, type 1 negation creates scope ambiguity.

Multiple negation is unacceptable in type 1 negation:

(50) * pi-ka ani ani onta.

(50) is ill-formed because the negative verb phrase ani onta is not basic. Note again that, in AMG, the notion of basic verb is more inclusive than the notion of lexical verb or that of basic expression in PTQ. In AMG, basic verbs include both lexically simple and complex verbs, some of which are derived by case shifting rules.[4]

5.3 Type 2 negation in Korean involves case marking.

(51) a. pi-ka o-či anihanta.
rain come Comp not
'It doesn't rain'

b. pi-ka o-či-lɨl anihanta
Acc

c. * pi-ka o-či-ka anihanta
Nom

In (51), the verb ending in *či* may be assigned a case. (51b) is acceptable, where the verbal complement is assigned the accusative case; while (51c) is unacceptable, where the complement is assigned the nominative case. On the other hand, we have the opposite pattern in (52):

(52) a. Mary-ka mip-či anihata
ugly Comp not
'Mary is not ugly'

b. Mary-ka mip-či-ka anihata
Nom

c. *Mary-ka mip-či-lɨl anihata
Acc

(52b) is acceptable, while (52c) is not.

This difference in case assignment may be due to the type of verb to which the case is assigned: an action verbal complement is assigned the accusative case, and a description verbal complement, the nominative case. But the following data indicate that what assigns a case is not the complementized verb phrase itself but the matrix verb that governs it.

(53) a. Mary-nɨn ča-l su-ka əpsta
Nom
Top sleep possible not
'Mary cannot sleep'

b. *Mary-nɨn ča-l su-lɨl əpsta

(54) a. Mary-nɨn ton-i əpsta
money Nom

'Mary does not have money'

b. *Mary-nɨn ton-ɨl əpsta

(54) shows that the matrix verb əpsta assigns cases to the noun phrases Mary and ton-money. So it should be the matrix verb əpsta that assigns cases to the noun phrase Mary and also to the complementized verb phrase ča-l su.

Likewise, in AMG, the negative verbs anihanta and anihata are treated as assigning cases to their verbal complements ending in či. For this, AMG simply subcategorizes anihanta and anihata as belonging to the categories $(10)_2 10$ and $(10)'_1 10$, respectively. On the basis of this subcategorization, we obtain the following well-formed analyses:

(55) [[pika]$_{+1}$[[[[o]$_{10}$-či]$_{(10)_n}$-lɨl]$_{(10)_2}$ [anihanta]$_{(10)_2 10}$]$_{10}$]$_{0}$

k1

k+1

(56) [[Maryka]$_{+1}$[[[[mip]$_{(10)'}$-či]$_{(10)'_n}$-ka]$_{(10)'_1}$ [anihata]$_{(10)'_1 10}$]$_{10}$]

k1

k+1

On the other hand, the following concatenations are blocked in AMG because they violate the categorial condition on concatenation:

(57) *pi-ka [[o-či-ka]$_{(10)_1}$ [anihanta]$_{(10)_2 10}$]$_{10}$

(58) *Mary-ka [[mip-či-lɨl]$_{(10)'_2}$ [anihata]$_{(10)'_1 10}$]$_{10}$

(59) *Mary-ka [[mip-či-ka]$_{(10)'_1}$ [anihanta]$_{(10)_2 10}$]$_{10}$

Like type 0 and 1 negations, type 2 negation also creates scope amibiguity.

(60) mot+n namča-ka o-či anihanta.

This has two readings: one means that no one comes, and the other means that not all come. To handle this, AMG subcategorizes the negative verb *anihanta* as of the categories $(10)_2 10$ and $(10)_2{+}10$, and the negative description verb *anihata* as of $(10)'_1 10$ and $(10)'_1{+}10$.

The negative verbs *anihanta* and *anihata* take as their argument not only a complementized IV-phrase but a verbal complement of any category. Thus the following analyses are all acceptable:

(61) a. Mary-ka [[John-+l salaŋha-či]$_{(10)_n}$ anihanta]$_{10}$

b. Mary-ka John-+l [[salaŋha-či]$_{(+210)_n}$ anihanta]$_{+210}$

(61b) then allows the following reordering:

(62) John-+l Mary-ka [salaŋha-či anihanta]$_{+210}$

In order to allow the concatenation of the negative verbs, especially *anihanta*, with any type of verbal complement, we treat these negative verbs as belonging to the category of the form $(X10)^i_n X'10$, where X and X' are strings of categories such that X' = X or X+, the superscript i is a (possibly null) string of primes, and n = 1, 2, 3, ... Note that this generalization makes it possible to interpret both the negative adverb *ani* and the negative verb *anihanta* or *anihata* as the semantically same functor expressions taking a verbal expression as an argument. These negative expressions *ani* and *anihanta* or *anihata* differ from each other only syntactically: the former functions as an adverb, while the latter functions as a matrix verb taking a verbal complement as one of its arguments.

Unlike the adverb *ani*, the negative verbs allow multiple negation.

(63) pi-ka ani o-či aniha-či anihanta

(64) Mary-ka mip-či-ka aniha-či-ka aniha-či-ka anihata

The following analysis shows how multiple negation and case assignment are obtained in AMG:

(65) $[[\text{pika}]_{+1}[[[[\text{ani o}]_{10}\text{-či-lɨl}]_{\underline{(10)_2}}\ [\text{aniha}]_{\underline{(10)_2}10}\ ]_{10}\text{-ci-lɨl}]_{\underline{(10)_2}}\ .$

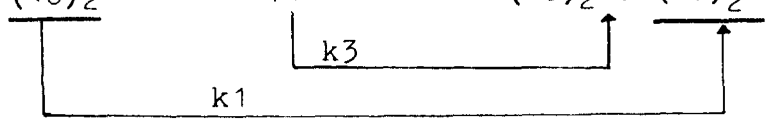
It can also shown easily that the following is ill-formed:

(66) *Maryka mipči-ka anihači-lɨl anihanta.

This is so because *mipči-ka anihači-lɨl* is of the category $(10)'_2$ and *anihanta*, of the category $(10)_2 10$. This shows that the case and any other relevant feature of the matrix verb, *anihanta* or *anihata*, percolate through the entire string of verbal complements.

As briefly noted in section 4, verbal complements may not be reordered.

(67) *[[o-či-lɨl]$_{(10)_2}$ [[pi-ka]$_{+1}$ [anihanta]$_{(10)_2 10}$]$_{(10)_2 0}$]$_0$



The concatenations k1 and k3 in (67) both satisfy the categorial condition, but (67) is ill-formed because the understood subject of the verbal complement *o-či-lɨl* is not controlled(?) by the subject *pi-ka* of the matrix verb. (But I will leave this issue for further study.)

5.4 So far I have shown how AMG analyzes three types of negation in Korean and how it solves the problems of case assignment, scope ambiguity, multiple negation, and reordering. For this purpose, AMG simply sets up three types of negative basic expressions:

(68) Basic Negative Expressions

a. *anita*: +110

+1+10

b. *ani*: (X10)X10

(X10)X+10 where X is a string of categories.

c. *anihanta*: $(X10)_2X10$

$(X10)_2X+10$

anihata: $(10)'_1 10$

$(10)'_1 +10$

AMG treats these negative expressions as semantically related functor expressions but as syntactically distinct: anita is a copular verb, ani an adverb, and anihanta or anihata a complement-taking verb. But note in particular that the adverb ani and the verb anihanta are exactly of the same semantic type, for the case index has no distinct semantic role. The copular verb anita and the complement-taking verb anihata are also related: they both require a double-nominative construction.

These negative expressions will assign proper cases to their argument expressions, for in AMG all the necessary information for case assignment is built into each functor expression and then each concatenation of a functor expression with its argument is constrained by the categorial condition:

(69) Categorial Condition on Concatenation

The concatneation $\alpha^\frown\beta$ or $\beta^\frown\alpha$ is well-formed if and only if α is a well-formed expression of the category A and β is that of the category (... A ...).

Secondly, the scope ambiguity of negative sentences, or possibly other opaque sentences, is accounted for by allowing negative expressions of the category X+10 that give rise to a wide scope reading.

Thirdly, multiple negation is blocked in type 1 negation because the negative adverb *ani* concatenates only with basic verbs including basic complex verbs. But it is allowed in type 2 negation because the case feature of the negative verb *anihanta* or *anihata* percolates through the entire string of verbal complements.

Lastly, reordering is allowed as long as it results from some process of concatenation that satisfies the categorial condition. But nothing can be inserted between the negative verb and its verbal complement, the reason of which still need be further studied.

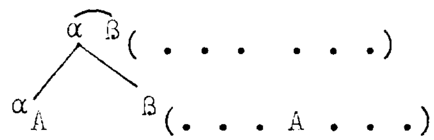
6. In concluding this paper, I like to make one remark on AMG: unlike PTQ, AMG successfully accounts for scope ambiguity without quantification rules. This alone may greatly reduce the power of Montague's PTQ and yet make it a descriptively adequate grammatical system.

Notes

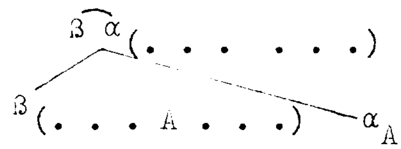
1. AMG employs several types of concatenation.

First, while k-concatenations concatenate α to the left of β, k'-concatenations concatenate α to the right of β, where α is of the category A and β, of the category (. . . A . . .).

k-concatenations:



k'-concatenations:



Secondly, concatenation k_i or k'_i concatenates α and β, if α is of the category A and β, of the category (... A ...), where A is the i-th place element in the sequence (... A ...).

Thirdly, concatenation k_{+i} or k'_{+i} concatenates α and β, if α is of the category +A and β, of the category (... A ...), where A is the i-th place element in the sequence (... A ...).

2. The negative verb anita is not ambiguous in an ordinary sense, but only in the sense that it is opaque as regards the scope of quantified phrases.

3. Besides analysis (45), AMG also allows the following analyses (a) and (b):

(a) $[[[\text{pi-ka}]_{+1}\ [\text{manhi}]_{(10)10}]_{(10)10}\ [\text{onta}]_{10}]_{0}$

k+2 k'1

(b) $[[\text{pi-ka}]_{+1}\ [[[\text{manhi}]_{(10)10}\ \ [\text{onta}]_{10}\ ]_{10}]_{0}$

k'1

k+1

These analyses claim that, in Korean, the noun phrase like pi-ka may freely be concatenated with the quantifier adverb manhi. But at present I am not sure whether this should be the case.

References

Dowty, David. 1979. Word meaning and Montague grammar. Dordrech: D. Reidel.

Kim, Gil Chang & Key Sun Choi. 1983. Parsing Korean based on Revised PTQ. (in this volume)

Lee, Kiyong. 1981a. On free word order and case in Korean: a Montague grammar approach (in Korean). Mal: a Linguistics Journal 6.51-87. Seoul: Yonsei University.

Lee, Kiyong. 1981b. Case assignment in Korean. Ŏhakgyoyuk Language Teaching 12.269-285. Gwangju, Korea: Chonnam National University.

Lee, Kiyong. 1983a. Case assignment and complex verbs in Japanese: a categorial approach. Festschrift for Prof. Masatake Muraki, ICU, Tokyo (in preparation).

Lee, Kiyong. 1983b. A Montague grammar for case languages. Proceedings of the XIIIth International Congress of Linguistics. (in preparation).

Montague, Richard. 1974. Formal Philosophy, ed. by Richmond Thomason. New Haven: Yale University.