20

Hungarian verb movement. A deductive perspective

Willemijn Vermaat

Utrecht University, Cognitive Artificial Intelligence
Enny Vredelaan 345, 3584 ZM Utrecht
vermaat@let.uu.nl

20.1 Introduction

As indicated by many researchers (Lecomte, 1998; Cornell, 1997) the minimalist framework as worked out in the Minimalist Program (Chomsky, 1995) and the resource conscious logics such as Multimodal Categorial Grammar (Moortgat, 1996) show many similarities. The two theories are the product of distinct scientific traditions, but they aim to give a similar explanation for grammatical knowledge and the use of that knowledge.

Stabler (1999) gives an algebraic framework to capture the main features of the Minimalist Program. Stabler's Minimalist Grammar (= MG) defines lexical feature specifications and tree structures to formulate grammars for Natural Language. A grammar consists of a lexicon, a set of feature specifications, closed under a set of basic structure building operations such as MERGE and MOVE.

Multimodal Categorial Grammar (MMCG) is a logical deductive theory of Natural Language. A multimodal grammar consists of a lexicon, logical formulas assigned to words, and operations, the rules of inferences. These rules are split up into logical inferences, the introduction and elimination rules of the unary and binary connectives, and structural inferences, which manipulate the structural composition between words.

In this paper, I will show how MMCG deals with the minimalist theory of movement in a specific case such as Hungarian verb movement. First I will specify the basic structure building operations MERGE and MOVE and show how they are defined in the Minimalist Grammar formalism of Stabler (1999). Section 20.3 describes the phenomenon of verb modifier climbing and

Proceedings of the ESSLLI Student Session 1999
Amalia Todirascu (editor).

shows how it is dealt with in the Minimalist Grammar approach. Section 20.4 gives three approaches how to handle verb modifier climbing in MMCG. In a broader perspective this section shows three ways to define the operation MOVE. In the conclusion I will show how the three approaches reveal that we should regard MOVE as a complex operation, which has both a structural and a logical component.

20.2 Structure building operations

20.2.1 Features

The structure building operations are applied to the lexical feature specifications. We distinguish phonological, semantic and syntactic features. In this paper we will abbreviate phonological and semantic features by writing the headword. The syntactic features are divided in two groups, which come in pairs: *category* features and *control* features. The category features state the role of a word in a sentence. Every word gets assigned a basic category feature $\mathcal{N}$, such as c for complementizer, v for verb or d for determiner. The role of a word is further determined by the selector features $=\mathcal{N}$. The selector feature indicates with what kind of category a word can be combined. The control features play an important role in controlling the order of words and the movement of phrases within structures. The licensee features $-\mathcal{N}$ state certain properties of words, such as $[-\texttt{wh}]$ or $[-\texttt{case}]$, while the licensor features $+\mathcal{N}$ indicate the need for such properties. The use of the two groups of features will be explained further in the definition of the two structure building operations MERGE and MOVE.

20.2.2 Merge

The Minimalist Grammar formalism (Stabler, 1999) defines MERGE as a structure building operation, which combines two tree structures t_1 the head of which carries a selector feature $[= \texttt{A}]$ and t_2 the head of which carries a corresponding category feature $[\texttt{A}]$. The operation MERGE causes the cancellation of the feature $[= \texttt{A}]$ against $[\texttt{A}]$. As tree t_1 can select both to the right and to the left, MERGE can be split into two operations[1]: $\text{MERGE}_<$ and $\text{MERGE}_>$. In MMCG MERGE is captured by modus ponens, which is defined in the natural deduction proof system by the elimination rules of the binary connectives $\{/, \backslash\}$. The next figure shows both structure building rules and the matching elimination rules: MERGE on the right as complement $[/_< E]$ and MERGE on the left as specifier $[\backslash_> E]$. Chomsky (1995) reasons about one MERGE operation, which can be regarded as the union of these two operations.

[1] In Stabler's Minimalist Grammar direction arrows $<$ and $>$ indicate the head of the tree.

$$\textsc{Merge}_{<}(t_1[=\mathtt{A}],t_2[\mathtt{A}]) \Rightarrow \overset{<}{\underset{t_1 \quad t_2}{\wedge}} \qquad \overset{>}{\underset{t_1 \quad t_2}{\wedge}} \Leftarrow \textsc{Merge}_{>}(t_2[=\mathtt{A}],\, t_1[\mathtt{A}])$$

$$\frac{t_1 \vdash B/_{<}A \quad t_2 \vdash A}{t_1 \circ_{<} t_2 \vdash B}\ [/_{<}E] \qquad \frac{t_1 \vdash A \quad t_2 \vdash A\backslash_{>}B}{t_1 \circ_{>} t_2 \vdash B}\ [\backslash_{>}E]$$

20.2.3 Move

MOVE is the mechanism which describes phenomena of "displacement"; lexical elements may appear at a different position than where they are interpreted. According to Chomsky (1995), uninterpretable features are responsible for this displacement. In the Minimalist Program he gives an explanation for the two linguistic notions *displacement* and *uninterpretability*. He links the two notions to build a theory on the operations MOVE/ATTRACT and the procedure *feature checking*.

Many lexical features are interpretable at the PF (phonological form) and the LF (logical form) interface, which means that the two interface levels need these features for interpretation. In contrast, some features are uninterpretable at the PF and/or the LF interface. These features have to be removed from the lexical structures by the procedure *feature checking*. Uninterpretable features have to be checked in a checking relation by features with corresponding feature values. Under influence of the operation ATTRACT the feature that needs to check an uninterpretable feature is attracted to the uninterpretable feature. The operation MOVE transfers the phrase that carries the matching feature to the position where it can check the uninterpretable feature.

Stabler (1999) defines MOVE as a structure building function which is presented by the following tree diagram. The 'uninterpretable' *licensor* feature $[+\mathtt{f}]$ on the head of the tree attracts the *licensee* feature $[-\mathtt{f}]$ on one of the subtrees $t_2{}^{>}$. After abstracting the subtree from the main tree, it is merged as a specifier to the head. The matching features are canceled and removed from the tree:

$$\overset{>}{\underset{\textsc{move}(t_1[+\mathtt{f}])=t_2{}^{>} \qquad t_1\{t_2[-\mathtt{f}]^{>}/-\}}{\wedge}}$$

In section 20.4, we show how we can capture the MOVE operation deductively.[2] The above definition already indicates that MOVE consists of two procedures *application* and *abstraction*. To capture these two procedures,

[2] A more technical approach to obtain the deductive meaning of MOVE can be read in Vermaat (1999)

the deductive approach of movement is split up as structural reasoning on the one hand and hypothetical reasoning on the other.

20.3 Verbal complex formation

Koopman and Szabolcsi (1998) explore the phenomenon of Hungarian verbal complexes. *Verbal complexes* appear in so-called neutral and non-neutral sentences. Neutral sentences are sentences without focused or negated subphrases. Verbal complexes are clusters of verbs, which are formed by two distinct *verbal complex formation* processes. The structures consist of verbal modifiers such as the prefix haza ('home'), auxiliaries such as akar ('want') and selecting verbs such as menni ('go').

The two verbal complex formation processes are *verb modifier climbing* (VM-climbing) and *verbal inversion*. The first process, which only occurs in neutral sentences, describes the climbing of the verbal modifier to precede the finite verb. The second process describes the recursive inversion of infinite verbs in non-neutral sentences with focus or negative phrases. This paper concentrates on VM-climbing in neutral sentences.

20.3.1 Verb modifier climbing

Verbal complex formation operates on sequences of auxiliaries. According to Koopman and Szabolcsi (1998), the underlying order of the auxiliary sequence in Hungarian is essentially the same as the English surface order. Under influence of a a process called verb modifier climbing (VM-climbing) the auxiliaries will form verbal complexes. In neutral sentences, the verbal complex is a sequence of infinitival auxiliaries combined with a finite auxiliary. The lowest infinitival selects a verbal modifier as complement. However, in neutral sentences the modifier precedes the finite auxiliary. VM-climbing causes the verbal modifier to procliticize to the finite verb.

The following examples show the order in which the auxiliaries occur in neutral sentences. In example (20.1) haza precedes the finite verb akarok. The basic 'English' order in example (20.2) is not allowed, as a prefix always appears in front of its selecting verb. Though the order in example (20.3) is not allowed either, the prefix is obliged to climb to the first position.

(20.1) haza akarok menni
home want[1sg] go[inf]
'I want to go home'

(20.2)* akarok menni haza (as neutral sentence)
want[1sg] go[inf] home

(20.3)* akarok haza menni (as neutral sentence)
want[1sg] home go[inf]

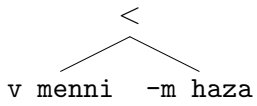
20.3.2 A Minimalist Grammar approach

In the Minimalist Program functional categories are responsible for the displacement of words. They carry uninterpretable features, which need to be checked. In the following MG grammar the licensor feature [+m] is specified on the functional category `I` and the licensee feature is part of the feature specification of the verbal modifier `haza` (= home). As an example, we will derive "`haza akarok menni`" in the Minimalist Grammar setting. Starting from lexical feature structures a binary tree structure is built.

(1) LEX: `m -m haza`

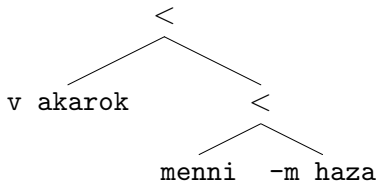
(2) LEX: `=m v menni`

(3) MERGE(2,1):



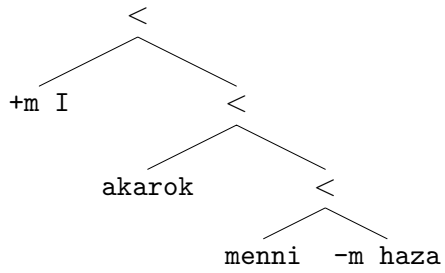
(4) LEX: `=v v akarok`

(5) MERGE(4,3):

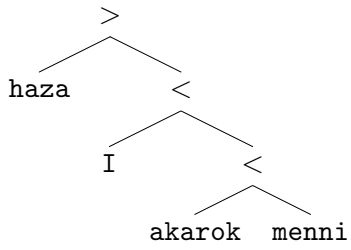


(6) LEX: `=v +m I`

(7) MERGE(6,5):



(8) MOVE(7):



20.4 A deductive approach

Stabler's formalism accounts for movement by marking the modifier with a licensee feature; the functional category carries the trigger, the corresponding licensor feature. The structure building operations direct the lexical feature structures to derive the right structural order. In MMCG, using the internal logical and structural reasoning facilities, the basic mechanisms can be split up in logical and structural operations. As we have already seen in section 20.2 MERGE is defined by modus ponens, the elimination rules of $\{/_<, \backslash_>\}$

MOVE has to be considered as an operation composed of a logical part and a structural part. The structural part captures the actual movement

of words through a sentence. In MMCG, postulates are used to control the structural composition of words. The use of these postulates is restricted by the unary connective $\langle . \rangle^f$, the structural counterpart of $\Diamond_f$ and $\Box_f$. These so-called control features contain the properties of words and have a similar use as the licensor and licensee features in the MG framework. The logical behavior of the unary connectives makes the interaction between the structural and logical part of a derivation possible. Their behavior is presented in natural deduction by the introduction and elimination rules of $\Diamond_f$ and $\Box_f$.

$$\frac{\Gamma \vdash \Box_f A}{\langle \Gamma \rangle^f \vdash A}\ [\Box_f E] \qquad \frac{\langle \Gamma \rangle^f \vdash A}{\Gamma \vdash \Box_f A}\ [\Box_f I]$$

$$\frac{\Gamma \vdash A}{\langle \Gamma \rangle^f \vdash \Diamond_f A}\ [\Diamond_f I] \qquad \frac{\Delta \vdash \Diamond_f A \quad \Gamma[\langle A \rangle^f] \vdash B}{\Gamma[\Delta] \vdash B}\ [\Diamond_f E]$$

By defining 'movement postulates', movement is captured within the structural domain. But this does not capture the meaning of movement as involving abstraction. With the use of hypothetical reasoning as defined by the introduction rules of the base logic, one is able to account for the abstraction of a hypothetical phrase out of a fully built structure:

$$\frac{\Gamma \circ_i x : B \vdash t : A}{\Gamma \vdash \lambda x.t : A/_i B}\ [/_i I] \qquad \frac{x : B \circ_i \Gamma \vdash t : A}{\Gamma \vdash \lambda x.t : B\backslash_i A}\ [\backslash_i I]$$

Now we are able to define MOVE in MMCG, where we use *structural reasoning* on the one hand and *higher order reasoning* on the other. I propose three different approaches to define VM-climbing in MMCG. The three approaches show how different aspects of MOVE such as the trigger of movement, the economy of derivations and the derivational meaning are worked out in MMCG. The following basic categories are involved: v_i for infinitival verbs, v_f for finite verbs and m for modifiers.

20.4.1 Move as Structural Reasoning

The different words involved in the phenomenon of VM-climbing have a certain categorial status within phrasal structures. The categories of the distinct lexical entries are determined by the role they play within sentences. The auxiliary akarok (= want) is a finite verb, which selects an infinitival verb as a complement. The infinitival verb menni (= go) takes a modifier as complement. The prefix haza (= home) is a modifier, which needs to precede the finite verb and therefore carries a special modifier feature. To account for the modifier feature as trigger for VM-climbing, a unary connective decorated with a 'modifier feature': $\Box_m$ is introduced. All the features and characteristics of the words are stated in the lexicon:

$$\mathsf{akarok} \vdash v_f/_< v_i \qquad \mathsf{menni} \vdash v_i/_< m \qquad \mathsf{haza} \vdash \Box_m m$$

How does VM-climbing get triggered? The derivation in Fig. 20.1 shows that the modifier carries the licensee feature: $\Box_m$; the goal formula $\Box_m v_f$ carries the matching licensor feature. Under influence of the structural counterpart of the unary connective, the modifier feature: $\langle . \rangle^m$, *movement* postulates are triggered. The postulates resolve the feature checking of the licensor feature against the licensee feature.

$$\begin{aligned} \Diamond_m B \bullet_> A &\rightarrow A \bullet_< \Diamond_m B && [Pm] \\ \Diamond_m (A \bullet_> B) &\rightarrow \Diamond_m A \bullet_> B && [K1m] \\ \Diamond_m B \bullet_> (A \bullet_< C) &\rightarrow A \bullet_< (\Diamond_m B \bullet_> C) && [Mm] \end{aligned}$$

To derive a sentence such as haza akarok menni, the basic "English order" is built first: akarok menni haza. The modifier is moved higher in the structure under influence of its modifier feature so that it precedes its selecting verb menni. This is not the final position; it has to move again, under influence of postulate **Mm**, to precede the finite verb. The category of the whole sentence has type $\Box_m v_f$. The sentence carries a modifier feature indicating that the verb phrase has undergone VM-climbing.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{akarok} \vdash v_f/_< v_i \quad \dfrac{\mathsf{menni} \vdash v_i/_< m \quad \dfrac{\mathsf{haza} \vdash \Box_m m}{\langle \mathsf{haza} \rangle^m \vdash m}\,[\Box_m E]}{\mathsf{menni} \circ_< \langle \mathsf{haza} \rangle^m \vdash v_i}\,[/_< E]}{\mathsf{akarok} \circ_< (\mathsf{menni} \circ_< \langle \mathsf{haza} \rangle^m) \vdash v_f}\,[/_< E]}{\mathsf{akarok} \circ_< (\langle \mathsf{haza} \rangle^m \circ_> \mathsf{menni}) \vdash v_f}\,[Pm]}{\langle \mathsf{haza} \rangle^m \circ_> (\mathsf{akarok} \circ_< \mathsf{menni}) \vdash v_f}\,[Mm]}{\langle \mathsf{haza} \circ_> (\mathsf{akarok} \circ_< \mathsf{menni}) \rangle^m \vdash v_f}\,[K1m]}{\mathsf{haza} \circ_> (\mathsf{akarok} \circ_< \mathsf{menni}) \vdash \Box_m v_f}\,[\Box_m I]$$

Figure 20.1: VM-climbing with structural reasoning

20.4.2 Move as abstraction

To invoke the abstraction of a hypothetical subphrase, the lexical formula of the word which undergoes VM-climbing, the prefix, needs to be changed. Haza is lifted to the higher order type: $v_f/_>(\Diamond_m \Box_m m \backslash_> v_f)$. Haza has to combine with a finite verb phrase from which a modifier has been abstracted. The logical formula assigned to haza contains both a licensor $\Diamond_m$ and a licensee $\Box_m$ feature. As a consequence, haza is the word that triggers VM-climbing; the modifier itself is responsible for the abstraction. Again, the structural counterpart of this 'lexical' licensor feature $\langle . \rangle^m$ triggers the movement postulates. Fig. 20.2 shows that the same postulates that were needed for the VM-climbing in the fragment with structural reasoning are needed here. The postulates move the hypothesized modifier out of its former position to the specifier position, from where it can be abstracted.

$$\dfrac{\mathsf{haza} \vdash v_f/_{>}(\Diamond_m\Box_m m\backslash_{>} v_f) \quad \dfrac{[\mathsf{r}_0 \vdash \Diamond_m\Box_m m]^1 \quad \dfrac{\mathsf{r}_0 \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \vdash v_f}{\mathsf{akarok} \circ_{<} \mathsf{menni} \vdash \Diamond_m\Box_m m\backslash_{>} v_f}\,[\backslash_{>}I]^1}{\langle \mathsf{r}_0 \rangle^m \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \vdash v_f}\,[\Diamond E]^2}{\mathsf{haza} \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \vdash v_f}\,[/_{>}E]$$

$$\dfrac{\mathsf{akarok} \vdash v_f/_{<}v_i \quad \dfrac{\mathsf{menni} \vdash v_i/_{<}m \quad \dfrac{[\mathsf{p}_1 \vdash \Box_m m]^2}{\langle \mathsf{p}_1 \rangle^m \vdash m}\,[\Box_m E]}{\mathsf{menni} \circ_{<} \langle \mathsf{p}_1 \rangle^m \vdash v_i}\,[/_{<}E]}{\mathsf{akarok} \circ_{<} (\mathsf{menni} \circ_{<} \langle \mathsf{p}_1 \rangle^m) \vdash v_f}\,[/_{<}E]$$

$$\dfrac{\mathsf{akarok} \circ_{<} (\langle \mathsf{p}_1 \rangle^m \circ_{>} \mathsf{menni}) \vdash v_f}{\langle \mathsf{p}_1 \rangle^m \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \vdash v_f}\,[Mm]$$

Figure 20.2: VM-climbing with hypothetical reasoning

20.4.3 Merge over Move

We can now improve the derivation by assigning haza the more economical type: $\mathsf{haza} \vdash \Box_m(v_i/_{>}(v_i/_{<}m))$. This type assignment accounts for MOVE as an abstraction operation. But it also captures the economy principle of 'MERGE over MOVE' where function application is favored over structural movement. This makes the derivation as presented in Fig. 20.3 more economical in the minimalistic sense.

$$\dfrac{\dfrac{\dfrac{\mathsf{haza} \vdash \Box_m(v_i/_{>}(v_i/_{<}m))}{\langle \mathsf{haza} \rangle^m \vdash v_i/_{>}(v_i/_{<}m)}\,[\Box_m E] \quad \mathsf{menni} \vdash v_i/_{<}m}{\langle \mathsf{haza} \rangle^m \circ_{>} \mathsf{menni} \vdash v_i}\,[/_{>}E] \quad \mathsf{akarok} \vdash v_f/_{<}v_i}{\mathsf{akarok} \circ_{<} (\langle \mathsf{haza} \rangle^m \circ_{>} \mathsf{menni}) \vdash v_f}\,[/_{<}E]$$

$$\dfrac{\dfrac{\dfrac{\mathsf{akarok} \circ_{<} (\langle \mathsf{haza} \rangle^m \circ_{>} \mathsf{menni}) \vdash v_f}{\langle \mathsf{haza} \rangle^m \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \vdash v_f}\,[Mm]}{\langle \mathsf{haza} \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \rangle^m \vdash v_f}\,[K1m]}{\mathsf{haza} \circ_{>} (\mathsf{akarok} \circ_{<} \mathsf{menni}) \vdash \Box_m v_f}\,[\Box_m I]$$

Figure 20.3: VM-climbing using higher order reasoning

The derivation in Fig. 20.3 slightly deviates from the derivation in Fig. 20.1 with respect to the order of the applications of words. Haza selects menni to form the substructure haza menni where menni is still regarded as the head of the structure. The derivation is more economical than the derivation in Fig. 20.1; less structural postulates are needed to derive the same structure.

20.5 Conclusion

The previous section shows three approaches to handle the phenomenon of VM-climbing, where logic, structure and derivational meaning each contribute to the characterization of the operation MOVE. In the first approach, the actual movement steps are captured with structural reasoning. The licensor features on the goal category project structural domains which trigger movement postulates. The second approach shows how the licensor features are defined on the 'moving' elements themselves to serve as triggers for movement. The interaction between licensee and licensor features on the logical and structural side eliminates the need for functional categories. The last approach accounts for the idea of Chomsky (1995) that minimal derivations obey certain economy principles, such as 'MERGE over MOVE'. To account for such a principle, we make use of the internal logic of higher order reasoning. A lexical category that is lifted to a higher order type makes it possible to reason hypothetically about MERGE.

As we have seen, the essence of the MOVE operation is captured within the multimodal framework. The goal of the Minimalist Program is to produce economical derivations. The MMCG framework may offer new insights in the discussion of economical derivations. What do the three approaches contribute to this discussion?

The three approaches show that there is no need for functional projections. The elimination of functional categories is a simplification for both the lexical and the derivational complexity. The lexicon only contains the necessary feature information for the derivation of sentences. One needs fewer lexical entries to get the same result. The derivational complexity is reduced by the assignment of higher order formulas to the moving elements. The second and third approach elaborate the idea that MOVE is not a primitive operation, but can be decomposed into abstraction and structural reasoning. By lifting the simple type formula of the moving element to a higher order type, we can reason hypothetically about MERGE. In this way one needs less structural rules to derive the same structure and therefore the derivational complexity reduces.

The third approach shows derivational improvement over the second approach, because it supports the economy condition 'MERGE over MOVE'. To be able to choose between the two approaches, the economical advantages of higher order reasoning versus the application of structural postulates should be further investigated. Some points of discussion lie in the differences between the placeholder of the licensor feature and the use of structural postulates ($[Pm]$ versus $[K1m]$). Therefore this paper raises an interesting question for further research: What aspects of MOVE make a derivation more economical?

Bibliography

Chomsky, N. (1995). *The Minimalist Program*, MIT Press, Cambridge, MA. Chapter 4.

Cornell, T. (1997). A type-logical perspective on minimalist derivations, *Formal Grammar 1997*, Aix-en-Provence, France. http://panza.sfs.nphil.uni-tuebingen.de/∼cornell/.

Koopman, H. and Szabolcsi, A. (1998). Verbal complexes, Draft.

Lecomte, A. (1998). Categorial minimalism, Forthcoming.

Moortgat, M. (1996). Categorial type logics, *in* J. van Benthem and A. ter Meulen (eds), *Handbook of Logic and Language*, Elsevier and MIT Press, Amsterdam and Cambridge MA, chapter 2, pp. 93–177.

Stabler, E. (1999). Remnant movement and structural complexity, *in* Bouma, Hinrichs, Kruijff and Oehrle (eds), *Constraints and Resources in Natural Language*, Syntax and Semantics. Forthcoming.

Vermaat, W. (1999). *Controlling movement: Minimalism in a deductive perspective*, Master's thesis, Utrecht University.

Proceedings of the ESSLLI Student Session 1999
Amalia Todirascu (editor).