# Solving Event Quantification and Free Variable Problems in Semantics for Minimalist Grammars

**Yu Tomita**
Department of Informatics, SOKENDAI
National Institute of Informatics
tomita@nii.ac.jp

## Abstract

In this paper, I will focus on the event quantification and free variable problems in semantics for Minimalist Grammar formalism, with which both Montagovian compositional semantics and neo-Davidsonian event semantics are compatible. In the literature, two operations of the formalism, MERGE and MOVE, are considered to correspond to semantic operations. MERGE corresponds to functional application or predicate conjunction, whereas MOVE determines a quantifier scope. In these semantic frameworks for Minimalist Grammar, however, the event quantification and free variable problems remain unsolved. Here I first propose a compositional event semantics for a subset of the formalism without MOVE and show that compositional event semantics is a key to solving the event quantification problem. Then, I extend the result to the formalism enriched with MOVE, and show that it is unnecessary to add any free variables or assignment functions.

## 1 Introduction

*Minimalist Grammars* (MGs), a grammar formalism which first appeared in (Stabler, 1997), have two operations, MERGE and MOVE[1]. The former combines two expressions into another expression. The latter extracts a subconstituent from a complex expression and remerges them. MGs are compatible with both Montagovian compositional semantics (Montague, 1976; Heim and Kratzer, 1998) and neo-Davidsonian event semantics (Parsons, 1990; Larson and Segal, 1995). In the literature, MERGE corresponds to functional application in Montagovian compositional semantics (Kobele, 2006, 2012) or predicate conjunction in neo-Davidsonian event semantics (Hunter, 2010a,b), respectively, whereas MOVE determines quantifier scope.

The previous studies (Hunter, 2010b; Kobele, 2006) assume the interaction between MOVE operation and quantification, which requires semantic expressions to contain free variables or assignment functions. The problems we have to consider here are the event quantification problem (Winter and Zwarts, 2011), which remains unanswered in (Hunter, 2010a), and the problem with free variables (Kobele, 2006).

This paper contains the following sections. In section 2, I introduce a tiny subset of MGs, and I will take up Montagovian and neo-Davidsonian semantic frameworks following (Hunter, 2010a). Then, in section 3, I will show that MOVE-free MG introduced in section 2 has the event quantification problem and I introduce a neo-Davidsonian logical form invented by (Champollion, 2015), which is compatible with pure functional application, showing that this framework can solve the event quantification problem. In section 4, I try to extend the proposal to full MG, considering the relationship between determination of quantifier scope and movement operation without adding any free variables or assignment functions.

[1] In this paper, I ignore the operation ADJOIN, which is added to MGs in (Frey and Gärtner, 2002; Hunter, 2010a,b), and should be considered in further studies.

## 2 MOVE-free MGs

In this section, we will look at MOVE-free MG and its semantics following (Hunter, 2010a). For the moment, we can ignore MOVE and its corresponding semantic operation.

Let $V$ be a finite set of *phonological expressions* (or *strings*) containing the empty string $\varepsilon$ and $B = \{\mathsf{c}, \mathsf{d}, \mathsf{n}, \mathsf{v}, ...\}$ be a set of *selection features*. $B$ (selectees) determines $B_{=} = \{=\mathsf{b} \mid \mathsf{b} \in B\}$ (selectors). Let $\{:, ::\}$ be the set of expression markers. In the following, we assume $s, t \in V^*$, $\circ_1, \circ_2 \in \{:, ::\}$, $\mathsf{g}, \mathsf{b}, \mathsf{b}_i \in B$ for $0 \leq i$.

A MOVE-free MG $G$ is a 4-tuple $\langle V, B, Lex, \mathsf{v}\rangle$ where $Lex$ is a finite subset of $V \times \{::\} \times (B_{=}^* \times B)$. Given $G$, the set of expressions $Expr$ is a union of the set of *lexical expressions*, namely, $Lex$ and a subset of $V^* \times \{:\} \times (B_{=}^* \times B)$. The latter expressions marked with ':' are called *complex expressions*. MERGE is an operation to derive a sentence. It combines two expressions into a complex expression by saturating =b with b. There are two cases of MERGE: the complement merge case (MERGE$_1$) and the specifier merge case (MERGE$_2$). If the expression carrying =b is indicated by '::', then MERGE$_1$ is applied; otherwise, MERGE$_2$ is applied.

(1) $$\frac{s :: =\mathsf{b}_1 ... =\mathsf{b}_n \mathsf{g} \qquad t \circ_1 \mathsf{b}_1}{st : =\mathsf{b}_2 ... =\mathsf{b}_n \mathsf{g}} \text{MERGE}_1$$

(2) $$\frac{s : =\mathsf{b}_1 ... =\mathsf{b}_n \mathsf{g} \qquad t \circ_1 \mathsf{b}_1}{ts : =\mathsf{b}_2 ... =\mathsf{b}_n \mathsf{g}} \text{MERGE}_2$$

A derivation is complete when the expression has the distinguished feature v only. An example derivation for *Brutus stabbed Caesar* is shown in Figure 1.

### 2.1 Semantics for the MOVE-free MG

Let us discuss semantics for a $G$ next. The sentence (3) has semantic representations as shown in (4). Montagovian compositional semantics (henceforth compositional semantics) yields the semantic representation (4a). On the other hand, neo-Davidsonian event semantics (henceforth event semantics) composes a more complicated logical form (4b).

(3) Brutus stabbed Caesar

(4) a. $\textbf{stab}(\textbf{b})(\textbf{c})$

b. $\exists e.\textbf{stabbing}(e) \wedge \textbf{stabbee}(\textbf{c})(e) \wedge \textbf{stabber}(\textbf{b})(e)$

We here call $\exists e$ an event quantifier. Note that this paper uses $e$ both as an event variable and as the type for individuals. An event variable $e$ has semantic type $v$, whereas every individual has type $e$. Let $\mathcal{S}$ be a set of *semantic expressions* (or *semantic components*). In the following sections, strings and expression markers are omitted in derivations.

#### 2.1.1 Compositional Semantics

In the compositional semantics, each lexical expression has a semantic component, as in the following entries:

(5)
Brutus::d, $\textbf{b}$
Caesar::d, $\textbf{c}$
stabbed::=d =d v, $\lambda xy.\textbf{stab}(x)(y)$

MERGE combines two semantic expressions. We here suppose that MERGE involves functional application of two semantic components. If $P$ and $Q$ are semantic components assigned to expressions, then the general scheme involved in MERGE is as shown in (6).

(6) $$\frac{=\mathsf{b}_1 ... =\mathsf{b}_n \mathsf{g}, P \qquad \mathsf{b}_1, Q}{=\mathsf{b}_2 ... =\mathsf{b}_n \mathsf{g}, P\,Q}$$

The truth condition for the semantic expression comes from itself.

#### 2.1.2 Event Semantics

Hunter (2010a) proposes an event semantic framework incorporated into MOVE-free MG. In this framework, each feature in a lexical item is annotated with semantic constants, as in the following entries:

(7)
Brutus::$\langle \mathsf{d}, \textbf{b}\rangle$
Caesar::$\langle \mathsf{d}, \textbf{c}\rangle$
stabbed::$\langle =\mathsf{d}, \textbf{stabbee}\rangle\langle =\mathsf{d}, \textbf{stabber}\rangle$
$\langle \mathsf{v}, \textbf{stabbing}\rangle$

We here call semantic expressions assigned to selectees *main predicates*, which have type $\langle v, t\rangle$, and those assigned to selectors *thematic relations*, which have type $\langle e, vt\rangle$.

The general schema involved in MERGE operation is shown in (8)

(8) $$\frac{\langle =\mathsf{b}_1, \theta_1\rangle ... \langle =\mathsf{b}_n, \theta_n\rangle\langle \mathsf{g}, P\rangle \qquad \langle \mathsf{b}_1, Q\rangle}{\langle =\mathsf{b}_2, \theta_2\rangle ... \langle =\mathsf{b}_n, \theta_n\rangle\langle \mathsf{g}, P\&\theta_1(Q)\rangle}$$

where $P\&R = \lambda e.P(e) \wedge R(e)$. We here call this compositional scheme *predicate conjunction*. If the

$$\dfrac{\dfrac{\texttt{stabbed::=d =d v} \qquad \texttt{Caesar::d}}{\texttt{stabbed Caesar:=d v}}\text{MERGE}_1 \qquad \texttt{Brutus::d}}{\texttt{Brutus stabbed Caesar:v}}\text{MERGE}_2$$

Figure 1: Derivation of *Brutus stabbed Caesar*

sentential expression denotes a predicate $P$, its type is $\langle v, t\rangle$ and its truth condition is $\exists e.[P(e)]$.

## 3 Event Quantification Problem

In this section, we will consider the event quantification problem in semantics for $G$. One of the incompatibilities between event semantics and compositional semantics comes from the fact that quantificational arguments obligatorily take scope over event quantifier, as suggested in (Fry, 2005). For example, a sentence (9) has a reading in (10a), but not (10b). In other words, since *nobody* is a quantifying expression $\lambda k.\neg\exists x.k\,x$, it cannot take scope under the event quantifier.

(9) Nobody walked

(10) a. $\neg\exists z.\exists e.\textbf{walking}(e) \wedge \textbf{walker}(z)(e)$

b. $\exists e.\neg\exists z.\textbf{walking}(e) \wedge \textbf{walker}(z)(e)$

In the compositional scheme of (Hunter, 2010a), however, quantifying determiner phrases merged into a verb phrase must take scope under the event quantifier. The question then arises: how can quantifying expressions such as *nobody* take scope over the event quantifier via the compositional scheme for event semantics?[2]

### 3.1 Compositional Event Semantics for MOVE-free MG

To avoid the event quantification problem, we here adopt the proposal in (Champollion, 2015) to MOVE-free MG. First, the main predicates contain an event quantifier and have a GQ type over events $\langle vt, t\rangle$, rather than $\langle v, t\rangle$. This means that sentences and verb phrases are assumed to take $\lambda e.\top$ to produce a proposition.

(11) $P_{\textbf{doing}} = \lambda f.\exists e.f(e) \wedge \textbf{doing}(e)$

Second, a separate semantic component for giving thematic relations to an argument is assigned to each selector feature.

[2]A solution in Abstract Categorial Grammar is illustrated in (Winter and Zwarts, 2011).

(12) $\theta_{\textbf{r}} = \lambda MNf.[M(\lambda x.[N(\lambda e.[f(e) \wedge \textbf{r}(x)(e)])])]$

Here, **r** is a thematic relation constant. The overall semantic expressions assigned to each lexical item are shown in Table 1.

Then, we can compose the argument and main predicate via pure functional application without giving rise to the event quantification problem. MERGE checks features in two expressions, applying a semantic component assigned to the selector feature to an argument.

(13) $$\dfrac{\langle \texttt{=b}_1, \theta_{\textbf{r}1}\rangle \ldots \langle \texttt{=b}_n, \theta_{\textbf{r}n}\rangle\langle \texttt{g}, P\rangle \qquad \langle \texttt{b}_1, Q\rangle}{\langle \texttt{=b}_2, \theta_{\textbf{r}2}\rangle \ldots \langle \texttt{=b}_n, \theta_{\textbf{r}n}\rangle\langle \texttt{g}, (\theta_{\textbf{r}1}Q)P\rangle}$$

Here the arguments have GQ type $\langle et, t\rangle$. If the argument has only selectee feature, MERGE involves functional application of the semantic component $\theta_{\textbf{r}1}Q$ to the main predicate $P$. For example, the logical form for the sentence (9) is derived in the following way:

(14) $$\dfrac{\langle \texttt{=d}, \theta_{\textbf{walker}}\rangle\langle \texttt{v}, P_{\textbf{walking}}\rangle \qquad \langle \texttt{d}, Q_{\textbf{nobody}}\rangle}{\langle \texttt{v}, (\theta_{\textbf{walker}}\, Q_{\textbf{nobody}})\, P_{\textbf{walking}}\rangle}$$

where

$$\begin{aligned}\theta_{\textbf{walker}} &= \lambda MNf.[M(\lambda x.[N(\lambda e.[f(e)\wedge \textbf{walker}(x)(e)])])],\\ Q_{\textbf{nobody}} &= \lambda k.\neg\exists z.k\,z,\\ P_{\textbf{walking}} &= \lambda f.\exists e.f(e) \wedge \textbf{walking}(e),\end{aligned}$$

and

$$(\theta_{\textbf{walker}}\, Q_{\textbf{nobody}})\, P_{\textbf{walking}} = \lambda f.\neg\exists z.\exists e.f(e)\wedge \textbf{walking}(e)\wedge \textbf{walker}(z)(e).$$

When the sentential expression denotes a predicate $P$, its truth condition is $P(\lambda e.\top)$.

In summary, I considered the event quantification problem in MOVE-free MG. In the next section, I will expand the proposed idea into full MG enriched with MOVE.

Table 1: Examples of lexical expressions for a subset of MGs

| | |
|---|---|
| Brutus:: | $\langle$d, $\lambda k.k\,\mathbf{b}\rangle$ |
| Caesar:: | $\langle$d, $\lambda k.k\,\mathbf{c}\rangle$ |
| everyone:: | $\langle$d, $\lambda k.\forall x.k\,x\rangle$ |
| someone:: | $\langle$d, $\lambda k.\exists y.k\,y\rangle$ |
| nobody:: | $\langle$d, $\lambda k.\neg\exists z.k\,z\rangle$ |
| walked:: | $\langle$=d, $\lambda MNf.[M(\lambda x.[N(\lambda e.[f(e) \wedge \mathbf{walker}(x)(e)])])]\rangle\langle$v, $\lambda f.\exists e.f(e) \wedge \mathbf{walking}(e)\rangle$ |
| stabbed:: | $\langle$=d, $\lambda MNf.[M(\lambda x.[N(\lambda e.[f(e) \wedge \mathbf{stabbee}(x)(e)])])]\rangle$ |
| | $\langle$=d, $\lambda MNf.[M(\lambda x.[N(\lambda e.[f(e) \wedge \mathbf{stabber}(x)(e)])])]\rangle\langle$v, $\lambda f.\exists e.f(e) \wedge \mathbf{stabbing}(e)\rangle$ |

## 4 Full MG and the Problem with Free Variables

The question we have to consider next is a problem with free variables. In the literature, MOVE operations in MGs require free variables or assignment functions in a semantic system. But the semantic components containing free variables or using assignment functions can cause some problems (Jacobson, 2015; Kobele, 2006). In this paper, however, we will observe that compositional event framework does not require any free variable or assignment function for MOVE.

### 4.1 A Minimalist Grammar Enriched with MOVE

Let $F$ = {o, q, ...} be a set of *licensing features*, where $F$ and $B$ are disjoint sets. $F$ determines $F_+ = \{+\mathtt{f} \mid \mathtt{f} \in F\}$ (licensors) and $F_- = \{-\mathtt{f} \mid \mathtt{f} \in F\}$ (licensees). Let Syn be the set of *feature bundles* $(B_= \cup F_+)^* \times B \times (F_-)^*$. We assume $\mathtt{f} \in F$, $\phi \in (B_= \cup F_+)^*$ , $\chi \in (B \times F_-^*)$, $\psi \in F_-^+$ and $\alpha_i, \beta_i \in (V^* \times F_-^+)$ for $0 \leq i$. For instance, $+\mathtt{f}\phi\chi, \mathtt{b}\psi \in$ Syn.

A *Minimalist Grammar* $G$ is a 5-tuple $\langle V, B, F, Lex, \mathtt{c}\rangle$ where $Lex$ is a finite subset of $V \times \{::\} \times$ Syn. Given $G$, the set of expressions $Expr$ is a union of $Lex$ and a subset of $(V^* \times \{:\} \times \text{Syn}) \times (V^* \times F_-^+)^*$. A complex expression can contain a sequence of subconstituents, each of which is a tuple of sequences of strings and licensees. There are two operations to derive a sentence: MERGE and MOVE. Each operation is defined in Figure 2.

MERGE$_1$ and MERGE$_2$ are essentially same as in MOVE-free MG we discussed earlier, except that the subconstituents can occur in each expression. The nonfinal merge case (MERGE$_3$) takes two expressions into one expression with an additional subconstituent, without concatenation of strings in expressions. The last two cases of MERGE (MERGE$_4$ and MERGE$_5$) realize covert movement. In these cases, where a feature bundle contains a distinguished licensee feature -q, the way of concatenation of strings is the same as MERGE$_1$ and MERGE$_2$, leaving a phonologically vacuous subconstituent for covert movement[3].

A specifier move case (MOVE$_1$) applies to an expression which contains a licensing feature +f and a subconstituent with an unchecked feature -f and concatenates two strings in one expression, whereas a nonfinal move case (MOVE$_2$) only checks their features.

A derivation is complete when the expression has the distinguished feature c only and involves no subconstituent. Additional examples of lexical expressions are shown in Table 2.

### 4.2 Variable-Free Semantics for full MG

I may now proceed to semantics of full MG. In a similar fashion to MOVE-free MG, a semantic component is assigned to each selector =b and selectee (followed by a sequence of licensees) $\chi$. No semantic component is assigned to a licensor +f.

Additional lexical expressions for full MG include complementizer heads such as the one shown in (15).

(15) $\varepsilon :: \langle$=v, $\lambda Nf.N\,f\rangle\langle$c, $\lambda e.\top\rangle$

They have no thematic relation. Instead, complementizer heads provide $\lambda e.\top$ to make the logical form closed.

[3] Torr and Stabler (2016) have introduced essentially same operations for Directional Minimalist Grammar.

$$\frac{s :: \mathsf{=b}\phi\chi \qquad t \fatsemi_1 \mathsf{b}, \alpha_1 ... \alpha_m}{st : \phi\chi, \alpha_1 ... \alpha_m}\ \text{MERGE}_1 \qquad \frac{s : \mathsf{=b}\phi\chi, \alpha_1 ... \alpha_m \qquad t \fatsemi_1 \mathsf{b}, \beta_1 ... \beta_n}{ts : \phi\chi, \alpha_1 ... \alpha_m \beta_1 ... \beta_n}\ \text{MERGE}_2$$

$$\frac{s \fatsemi_1 \mathsf{=b}\phi\chi, \alpha_1 ... \alpha_m \qquad t \fatsemi_2 \mathsf{b}\psi, \beta_1 ... \beta_n}{s : \phi\chi, \alpha_1 ... \alpha_m (t, \psi) \beta_1 ... \beta_n}\ \text{MERGE}_3\ (\text{where } \psi \neq \mathsf{-q})$$

$$\frac{s :: \mathsf{=b}\phi\chi \qquad t \fatsemi_1 \mathsf{b}\ \mathsf{-q}, \alpha_1 ... \alpha_m}{st : \phi\chi, (\varepsilon, \mathsf{-q}) \alpha_1 ... \alpha_m}\ \text{MERGE}_4 \qquad \frac{s : \mathsf{=b}\phi\chi, \alpha_1 ... \alpha_m \qquad t \fatsemi_1 \mathsf{b}\ \mathsf{-q}, \beta_1 ... \beta_n}{ts : \phi\chi, \alpha_1 ... \alpha_m (\varepsilon, \mathsf{-q}) \beta_1 ... \beta_n}\ \text{MERGE}_5$$

$$\frac{s : \mathsf{+f}\phi\chi, \alpha_1 ... \alpha_i (t, \mathsf{-f}) \alpha_{i+1} ... \alpha_m}{ts : \phi\chi, \alpha_1 ... \alpha_i \alpha_{i+1} ... \alpha_m}\ \text{MOVE}_1 \qquad \frac{s : \mathsf{+f}\phi\chi, \alpha_1 ... \alpha_i (t, \mathsf{-f}\psi) \alpha_{i+1} ... \alpha_m}{s : \phi\chi, \alpha_1 ... \alpha_i (t, \psi) \alpha_{i+1} ... \alpha_m}\ \text{MOVE}_2$$

$\phi \in (B_{=} \cup F_{+})^*$; $\chi \in (B \times F_{-}^*)$; $\psi \in F_{-}^{+}$; $\alpha_i, \beta_i \in (V^* \times F_{-}^{+})$ for $0 \leq i$; $0 \leq m, n$

Figure 2: Operation for MGs enriched with MOVE

Table 2: Additional examples of lexical expressions for full MG

| |
|---|
| everyone::$\langle \mathsf{d}\ \mathsf{-q}, \lambda k.\forall x.k\, x \rangle$ |
| someone::$\langle \mathsf{d}\ \mathsf{-q}, \lambda k.\exists x.k\, x \rangle$ |
| man::$\langle \mathsf{n}, \lambda y.\mathbf{man}(y) \rangle$ |
| who::$\langle \mathsf{d}\ \mathsf{-o}, \lambda ky.k\, y \rangle$ |
| $\varepsilon$::$\langle \mathsf{=v}, \lambda N f.N\, f \rangle \langle \mathsf{c}, \lambda e.\top \rangle$ |
| $\varepsilon$::$\langle \mathsf{=v}, \lambda N f W.(W\, N)\, f \rangle \langle \mathsf{+o}, \emptyset \rangle \langle \mathsf{=n}, \lambda hky.(h\, y) \wedge (k\, y) \rangle \langle \mathsf{n}, \lambda e.\top \rangle$ |
| $\varepsilon$::$\langle \mathsf{=v}, \lambda N V.V\, N \rangle \langle \mathsf{+q}, \emptyset \rangle \langle \mathsf{v}, \emptyset \rangle$ |

The compositional scheme for full MG is as follows. $\text{MERGE}_3$, $\text{MERGE}_4$ and $\text{MERGE}_5$ involve functional application of an argument $Q$ and a semantic component assigned to selector $R$, but not a main predicate $P$. As a result, a moving subconstituent has a semantic representation $R\, Q$.

(16) $$\frac{\langle \mathsf{=b}, R \rangle \Phi \langle \chi, P \rangle,\ \Psi_1 \qquad \langle \mathsf{b}\psi, Q \rangle, \Psi_2}{\Phi \langle \chi, P \rangle, \Psi_1 \langle \psi, R\, Q \rangle \Psi_2}$$

Here, $\Phi \in ((B_{=} \times \mathcal{S}) \cup (F_{+} \times \{\emptyset\}))^*$ and $\Psi_1, \Psi_2 \in (F_{-}^{+} \times (\mathcal{S} \cup \{\emptyset\}))^*$. $\text{MOVE}_1$ checks licensing features, obligatorily applying functional application if a moving subconstituent contains a licensee `-f`.

(17) $$\frac{\langle \mathsf{+f}, \emptyset \rangle \Phi \langle \chi, P \rangle, \Psi_1 \langle \mathsf{-f}, Q \rangle \Psi_2}{\Phi \langle \chi, P\, Q \rangle, \Psi_1 \Psi_2}$$

If a moving subconstituent contains a sequence of licensee features, $\text{MOVE}_2$ is applied. In this case, functional application is optional.

(18) $$\frac{\langle \mathsf{+f}, \emptyset \rangle \Phi \langle \chi, P \rangle, \Psi_1 \langle \mathsf{-f}\psi, Q \rangle \Psi_2}{\Phi \langle \chi, P \rangle, \Psi_1 \langle \psi, Q \rangle \Psi_2}$$

(19) $$\frac{\langle \mathsf{+f}, \emptyset \rangle \Phi \langle \chi, P \rangle, \Psi_1 \langle \mathsf{-f}\psi, Q \rangle \Psi_2}{\Phi \langle \chi, P\, Q \rangle, \Psi_1 \langle \psi, \emptyset \rangle \Psi_2}$$

Here I suppose $P\, \emptyset = P$. I now discuss compositional event semantics for MOVE, such as wh-movement and Quantifier Raising.

### 4.2.1 Wh-Movement

In MG, a wh-phrase can move to the highest position of the clause which it first merged. For example, a nominal expression *man who everyone stabbed* is supposed to have the following structure:

(20) man [ who [ everyone stabbed ~~who~~ ] ]

Figure 3 shows the derivation of the expression *man who everyone stabbed*. Before *who* is moved to its surface position, phonetically null complementizer is merged with the clause to form the nominal expression[4].

[4] Since the semantic component of *who* has type $\langle et, et \rangle$, if $\theta_{\mathbf{r}}$ would have type $\langle \langle et, t \rangle, \langle \langle vt, t \rangle, \langle vt, t \rangle \rangle \rangle$, functional application of $\theta_{\mathbf{r}}$ with *who* causes type mismatch. Therefore, I suppose $\theta_{\mathbf{r}}$ has type $\langle \langle et, \gamma \rangle, \langle \langle vt, t \rangle, \langle vt, \gamma \rangle \rangle \rangle$, where $\gamma$ ranges over the semantic types.

(16) stabbed:: $\langle \texttt{=d}, \theta_{\textbf{stabbee}}\rangle\langle \texttt{=d}, \theta_{\textbf{stabber}}\rangle\langle \texttt{v}, P_{\textbf{stabbing}}\rangle$ who:: $\langle \texttt{d -o}, \lambda ky.k\,y\rangle$ — MERGE$_3$

stabbed: $\langle \texttt{=d}, \theta_{\textbf{stabber}}\rangle\langle \texttt{v}, P_{\textbf{stabbing}}\rangle, \langle \texttt{who, -o}, \lambda Nfy.[N(\lambda e.[f(e) \wedge \textbf{stabbee}(y)(e)])]\rangle$

(13) stabbed: $\langle \texttt{=d}, \theta_{\textbf{stabber}}\rangle\langle \texttt{v}, P_{\textbf{stabbing}}\rangle, \langle \texttt{who, -o}, \lambda Nfy.[N(\lambda e.[f(e) \wedge \textbf{stabbee}(y)(e)])]\rangle$ everyone:: $\langle \texttt{d}, \lambda k.\forall x.k\,x\rangle$ — MERGE$_2$

everyone stabbed: $\langle \texttt{v}, \lambda f.\forall x.\exists e.f(e) \wedge \textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e)\rangle, \langle \texttt{who, -o}, \lambda Nfy.[N(\lambda e.[f(e) \wedge \textbf{stabbee}(y)(e)])]\rangle$

(13) $\varepsilon$:: $\langle \texttt{=v}, \lambda NfW.(W N) f\rangle\langle \texttt{+o}, \emptyset\rangle\langle \texttt{=n}, \lambda hky.(h\,y) \wedge (k\,y)\rangle\langle \texttt{n}, \lambda e.\top\rangle$ everyone stabbed: $\langle \texttt{v}, \lambda f.\forall x.\exists e.f(e) \wedge \textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e)\rangle, \langle \texttt{who, -o}, \lambda Nfy.[N(\lambda e.[f(e) \wedge \textbf{stabbee}(y)(e)])]\rangle$ — MERGE$_1$

everyone stabbed: $\langle \texttt{+o}, \emptyset\rangle\langle \texttt{=n}, \lambda hky.(h\,y) \wedge (k\,y)\rangle\langle \texttt{n}, \lambda W.(W(\lambda f.\forall x.\exists e.f(e) \wedge \textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e)))\lambda e.\top\rangle, \langle \texttt{who, -o}, \lambda Nfy.[N(\lambda e.[f(e) \wedge \textbf{stabbee}(y)(e)])]\rangle$

(17) everyone stabbed: $\langle \texttt{+o}, \emptyset\rangle\langle \texttt{=n}, \lambda hky.(h\,y) \wedge (k\,y)\rangle\langle \texttt{n}, \lambda W.(W(\lambda f.\forall x.\exists e.f(e) \wedge \textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e)))\lambda e.\top\rangle, \langle \texttt{who, -o}, \lambda Nfy.[N(\lambda e.[f(e) \wedge \textbf{stabbee}(y)(e)])]\rangle$ — MOVE$_1$

who everyone stabbed: $\langle \texttt{=n}, \lambda hky.(h\,y) \wedge (k\,y)\rangle\langle \texttt{n}, \lambda y.\forall x.\exists e. \wedge \textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e) \wedge \textbf{stabbee}(y)(e)\rangle$

(13) who everyone stabbed: $\langle \texttt{=n}, \lambda hky.(h\,y) \wedge (k\,y)\rangle\langle \texttt{n}, \lambda y.\forall x.\exists e. \wedge \textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e) \wedge \textbf{stabbee}(y)(e)\rangle$ man:: $\langle \texttt{n}, \lambda y.\textbf{man}(y)\rangle$ — MERGE$_2$

man who everyone stabbed: $\langle \texttt{n}, \lambda y.\textbf{man}(y) \wedge \forall x.\exists e.\textbf{stabbing}(e) \wedge \textbf{stabber}(x)(e) \wedge \textbf{stabbee}(y)(e)\rangle$

Figure 3: Derivation of *man who everyone stabbed*

### 4.2.2 Quantifier Raising

Since (May, 1977), quantificational determiner phrases such as *everyone* and *someone* are supposed to be raised as if they were wh-phrases. This Quantifier Raising (QR) is different from wh-movement in terms of QR does not affect word order. Moreover, if a sentence contains more than one quantifier, it can have semantic ambiguity due to QR. For instance, the sentence (21) has two readings, surface scope reading (22a) and inverse scope reading (22b).

(21) Someone stabbed everyone

(22) a. $\exists y.\forall x.\exists e.\textbf{stabbing}(e) \wedge \textbf{stabbee}(x)(e) \wedge \textbf{stabber}(y)(e)$

b. $\forall x.\exists y.\exists e.\textbf{stabbing}(e) \wedge \textbf{stabbee}(x)(e) \wedge \textbf{stabber}(y)(e)$

In the proposed framework, both readings are derivable with additional entries shown in (23).

(23)
everyone:: $\langle \texttt{d -q}, \lambda k.\forall x.k\,x\rangle$
someone:: $\langle \texttt{d -q}, \lambda k.\exists x.k\,x\rangle$
$\varepsilon$:: $\langle \texttt{=v}, \lambda NV.V\,N\rangle\langle \texttt{+q}, \emptyset\rangle\langle \texttt{v}, \emptyset\rangle$

Figure 4 shows the derivation of *someone stabbed everyone* with inverse scope reading. Here I adopt the QR analysis. All quantificational determiner phrases covertly move to higher positions of the sentence than the positions where the determiners are merged, without changing word order. The proposed idea is similar to the *delta QP* analysis in (Ikuta, 2015). In his analysis, a phonologically vacuous head $\Delta$ has two features [+topic] and [+focus] to invoke movement of arguments, such as QR and topic/focus movement. Here I suppose that each phonologically vacuous head carries a licensor feature +q only.

## 5 Comparison with Other Approaches

There are some previous works investigating MOVE and semantic operation. The comparison of my proposal with (Kobele, 2012; Hunter, 2010b) is summarized in Table 3.

In (Hunter, 2010a), typical quantificational determiner phrases like *everyone* do not actually appear. Hunter (2010b) proposes Insertion Minimalist Grammar and a QR analysis which can solve the

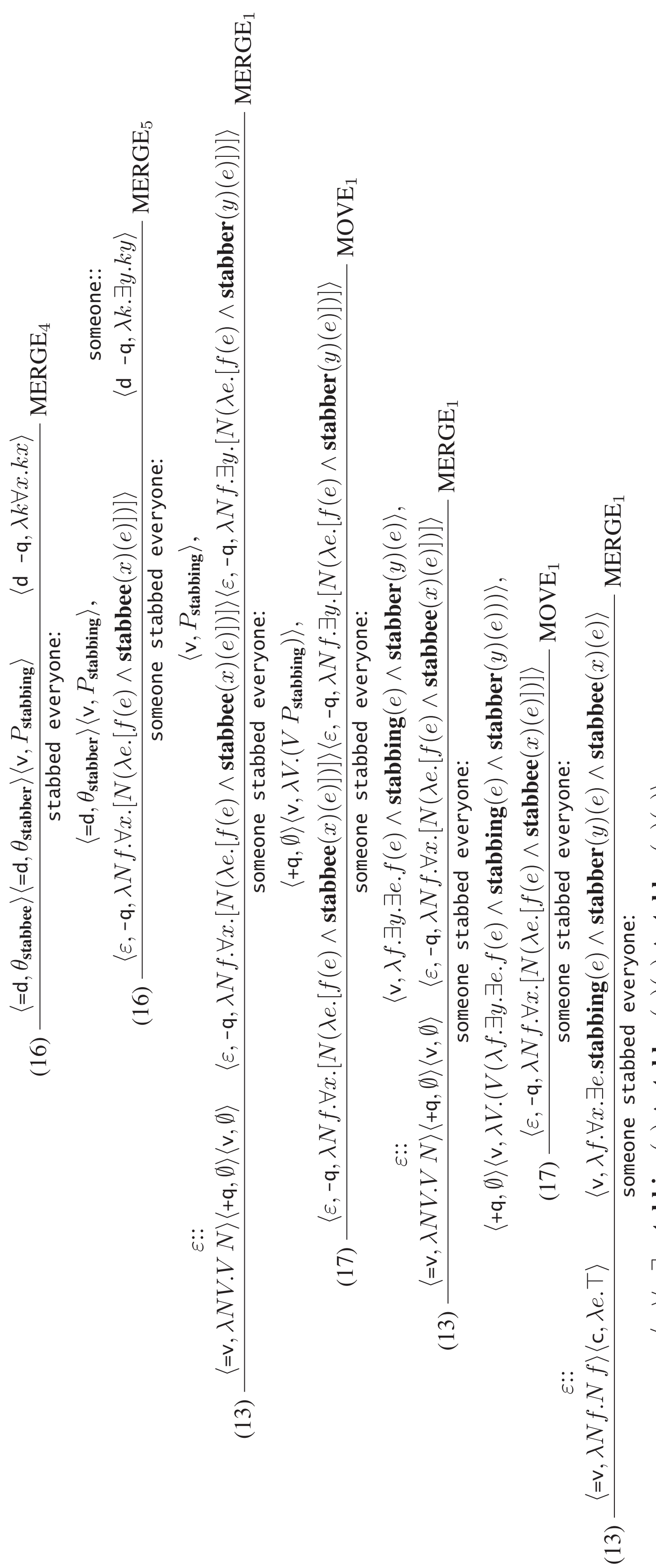

Figure 4: Derivation of *someone stabbed everyone*

Table 3: Comparison of semantic frameworks for MGs

| | Event quantification problem | Problem with free variable |
|---|---|---|
| (Kobele, 2012) | (No event) | ✓ |
| (Hunter, 2010b) | ✓ | × |
| My proposal | ✓ | ✓ |

event quantification problem. The problem with free variables, however, still remains.

A proposal in (Kobele, 2012) gives an interesting way to eliminate free variables in MG semantics. This paper shows that a compositional scheme (pure functional application) does not require any free variable or assignment function in MG semantics, due to just the event system which is not considered in (Kobele, 2012).

Besides those works on MGs, there are a few studies considering displacement operation and compositional semantics without free variables. Unger (2010) provides a variable-free semantics for a grammar formalism involving displacement operation similar to MOVE. She assumes that there is no correspondence between displacement and semantic operation. In contrast, I argue that MOVE corresponds to determination of a quantifier scope.

## 6 Conclusions

In this paper, I have illustrated a combination of compositional semantics and event semantics in MG formalism, showing that the compositional event semantic framework is the key to solve the event quantification problem. Moreover, I have given variable-free semantics for MG enriched with MOVE within the proposed scheme. This approach is different from any other previous works.

There is room for reconsidering compositional scheme and the semantic components to construct a complex sentence. The most important limitation lies in extraction from embedded clauses.

(24) a. Brutus saw a man who Caesar stabbed

b. Brutus saw a man who Longus thought Caesar stabbed

In the case of (24a), the meaning of an relative clause is derivable in the way I have shown in section 4.2.1. In the case of (24b), however, *who* moves out of two embedded clauses. If functional application of *who* to *Longus thought Caesar stabbed* is applied, then the thematic relation assigned to *who* might take the event variable associated with *thought*, not with *stabbed*. One way that we can think of to solve this puzzle is (i) successive-cyclic movement of *who* to *Caesar stabbed* involving functional application, and then (ii) employment of the thematic relation assigner for *thought* shown in (25), which is similar to the lifted thematic relation assigner proposed in (Champollion, 2015).

(25) $\theta_{\textbf{thinkee}} = \lambda W N V f.[W(\lambda e.[(V\ N)(\lambda o.[f(o) \wedge \textbf{thinkee}(e)(o)])])]$

(i) allows the semantic components for *who* to take the event variable associated with *stabbed*, not with *thought*. Then, inverse scope between two embedded clauses is derivable due to (ii) and the following lexical expressions.

(26)
$$\begin{array}{l}\textsf{who}{::}\langle \textsf{d -o -o}, \lambda k y.k\, y\rangle \\ \varepsilon{::}\langle \textsf{=v}, \lambda N U.U\ N\rangle\langle \textsf{+o}, \emptyset\rangle\langle \textsf{d}, \emptyset\rangle \\ \varepsilon{::}\langle \textsf{=v}, \lambda N f.(N\ f)\ \lambda e.\top\rangle\langle \textsf{+o}, \emptyset\rangle \\ \langle \textsf{=n}, \lambda h k y.(h\, y) \wedge (k\, y)\rangle\langle \textsf{n}, \lambda o.\top\rangle\end{array}$$

However, there is a more complicated puzzle shown in (27) which is unsolvable with (i) and (ii).

(27) Brutus met a man who everyone thought nobody stabbed

Though *everyone* must take scope over *nobody*, this solution requires only inverse scope reading.

In conclusion, the current study is unable to treat some embedded clauses and quantifying expressions properly. Clearly, however, there may be other possible explanation of a multiple embedded clause. Additional works on the compositional event semantics would be helpful to solve this problem. This approach has the potential to account more semantic descriptions than those I have shown in this paper.

## Acknowledgements

I appreciate Makoto Kanazawa and Hitomi Hirayama giving me helpful comments on the earlier versions of the present paper. I also thank the anonymous PACLIC reviewers for their comments. The earlier version of the paper was encouraged by comments offered at NII Logic Seminar. All errors are mine.

## References

Champollion, Lucas. 2015. The interaction of compositional semantics and event semantics. *Linguistics and Philosophy* 38:31–66.

Frey, Werner, and Hans-Martin Gärtner. 2002. On the Treatment of Scrambling and Adjunction in Minimalist Grammars. In *Proceedings of formal grammar*, ed. Gerhard Jäger, Paola Monacheshi, and Gerald Penn, 41–52.

Fry, John. 2005. Resource-logical Event Semantics for LFG (Draft). Unpublished Manuscript.

Heim, Irene, and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

Hunter, Tim. 2010a. Deriving Syntactic Properties of Arguments and Adjuncts from Neo-Davidsonian Semantics. In *The Mathematics of Language*, ed. Christian Ebert, Gerhard Jäger, and Jens Michaelis, volume 6149, 103–116. Springer Berlin Heidelberg.

Hunter, Tim. 2010b. Relating movement and adjunction in syntax and semantics. Doctoral Dissertation, University of Maryland.

Ikuta, Toshikazu. 2015. Interactions between Quantifier Scope and Topic / Focus. In *Proceedings of the Florida Linguistics Yearly Meeting (FLYM) 2*.

Jacobson, Pauline. 2015. *Compositional Semantics*. Oxford: Oxford University Press.

Kobele, Gregory Michael. 2006. Generating Copies : An investigation into structural identity in language and grammar. Doctoral Dissertation, University of California, Los Angels.

Kobele, Gregory Michael. 2012. Importing Montagovian Dynamics into Minimalism. In *Logical Aspects of Computational Linguistics*, ed. Denis Béchet and Alexander Dikovsky, 103–118. Springer Berlin Heidelberg.

Larson, Richard, and Gabriel Segal. 1995. *Knowledge of Meaning*. Cambridge, MA: MIT Press.

May, Robert. 1977. The Grammar of Quantification. Doctoral Dissertation, Massachusetts Institute of Technology.

Montague, Richard. 1976. *Formal Philosophy, Selected Papers of Richard Montague*. New Haven: Yale University Press.

Parsons, Terence. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. Cambridge, MA: MIT Press.

Stabler, Edward P. 1997. Derivational Minimalism. In *Logical Aspects of Computational Linguistics*, ed. Christian Retoré, 68–95. London, UK: Springer-Verlag.

Torr, John, and Edward P Stabler. 2016. Coordination in Minimalist Grammars : Excorporation and Across the Board (Head) Movement. In *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+12)*, 1–17.

Unger, Christina. 2010. A computational approach to the syntax of displacement and the semantics of scope. Doctoral Dissertation, Universiteit Utrecht.

Winter, Yoad, and Joost Zwarts. 2011. Event Semantics and Abstract Categorial Grammar. In *The Mathematics of Language*, ed. Makoto Kanazawa, András Kornai, Marcus Kracht, and Hiroyuki Seki, volume 6878, 174–191.