Glue 意味論の可能性

三藤　博

1.　はじめに

筆者は以前、イベント意味論の枠組みの中で、統語構造からイベント意味論的意味表示へと至る過程を、イベント意味論の意味表示の連言的(conjunctive)構成を利用する形で、順次連言的に組み合わせていく過程として構成することを提案した（三藤(2007)）が、この提案はセミフォーマルなものに留まり、厳密な形式的規則として書き下すには至っていなかった。他方、近年特に LFG(Lexical-Functional Grammar)や HPSG(Head-driven Phrase Structure Grammar)における意味論の一つの候補として、Glue 意味論という意味論が構成され、具体的な意味分析にも活用されてきている。Glue 意味論は、統語構造を意味論的論証への入力としてとらえ、逐次的に処理していくという点において、筆者が以前構想していた意味論的メカニズムを厳密に形式的に構成したものと言える。Glue 意味論は、これまで Minimalist 統語論に対しては構築されてこなかったが、最近 Gotham(2018)が Glue 意味論を Minimalist 統語論に対する意味論として具体的に構成し、Glue 意味論が Montague 以来の伝統的な形式意味論に比べて優れている点を有することを主張している。

そこで本稿では、Gotham(2018)に基づいて、Minimalist 統語論に対する意味論としての Glue 意味論の可能性について考察してみたい。

Glue 意味論は線形論理(linear logic)に立脚する意味論であるが、線形論理は従来の Montague 以来の伝統的な形式意味論ではほとんど全く用いられておらず、形式意味論研究者の間でのなじみも極めて薄いことと思われるため、まず次節において線形論理について簡潔に概観することから始めたい。

2.　線形論理(linear logic)

線形論理(linear logic)は、1980 年代の後半にフランス人論理学者 Jean-Yves Girard によって提唱された論理体系であり、比較的新しい論理体系であると言える。

線形論理の最大の特徴は、構成的な論理体系を実現することを徹底的に追求している点である。排中律（$A \vee \neg A$）の成立を否定する直観論理(intuitionistic logic)も構成的な論理であるとされるが、線形論理は直観論理にもまして構成性を追求している。この点を具体的に見れば、線形論理の推論規則においては、古典論理では当然認められるコントラクション(contraction)（(1)に示す）と弱化(weakening)（(2)に示す）がともに認められない。

(1) $\dfrac{\Gamma, A, A \mid\!- \ B}{\Gamma, A \mid\!- \ B}$

(2) $\dfrac{\Gamma \ \mid\!- \ B}{\Gamma, A \mid\!- \ B}$

弱化が認められないことから、古典論理では自明の如く成立する次の(3)の推論も許されないこととなる。

(3) $A, B \mid\!- \ A$

古典論理においては、(3) は $A \mid\!- \ A$ を始式(initial sequence)として、これに(2)の弱化を適用して直ちに得られるが、弱化が認められない線形論理ではこれができないわけである。

線形論理では(3)が成り立たないことを別の観点から見ると、線形論理では、前提を言わば「使い残す」形となる推論は認められないということになる。実際このことは線形論理の大きな特徴であり、このことから線形論理はしばしば「リソースの論理(logic of resources)」と呼ばれている。

(3)のように前提 B が「使われない」ような形の推論が許されないのに対して、線形論理においても古典論理における Modus Ponens は成立する。すなわち、(4)のような推論形式である。

(4) $\dfrac{A \ -\!\bigcirc \ B, \quad A}{B}$[1]

このように、推論の進行に際して必要にして十分な数の前提だけが認められるという線形論理の特徴は、自然言語の意味論のベースとしても優れた点であると言える。生成文法の流れの中でも Principle of Full Interpretation ということが提起されたことからも明らかなとおり、また形式意味論における構成性の原理(the principle of compositionality)が統語論と意味論の間に準同型関係(homomorphism)を要請していると捉えると、ある一つの syntactic object の意味への寄与は「1回限り」であるはずであり、同一の syntactic object が意味部門に対して複数個の入力となるといった事態は考えられないわけであるが、このことは線形論理が持つ上記の性質（推論の進行に際して必要にして十分な数の前提だけが

---

[1] 「－○」は本来一体の記号で、線形論理における含意(implication)を表す。本来は「○」の部分を小さくして「－」に繋げて一体化するべきなのであるが、本稿における作成の都合上、このような形式による表示としている。以下すべて同じである。

認められるという性質）とよく合致しており、線形論理をベースとした自然言語意味論の可能性が当然のごとく浮かび上がってこよう。そして、まさにその構想を実現したものがGlue 意味論にほかならない。

そこで次節では、Gotham(2018)に基づいてGlue意味論について概観してみることとしよう。

3. Glue 意味論

Glue 意味論は、前節で概観した線形論理の特徴を活かして、意味解釈のプロセスを線形論理における証明(proof)と見なして、前節でも少し述べたように、証明のための前提が統合論から意味論への入力として供給される、という構成を取る。

この構想を具体化するにはいくつかの可能性があるが、本稿では Gotham(2018)が採用しているメカニズムに基づいて考察を進めていきたい。このメカニズムは Glue 意味論として標準的なものと言って差し支えないであろう。

まず、統語論から供給された各々の前提が、「意味構成子(meaning constructor)」と呼ばれるペアの形に変換される。このペアは、λ-計算の表式(expression)と線形論理の論理式を組み合わせたもので、次の(5)のような形式で表示される。

(5)　意味構成子(meaning constructor)

m：ϕ　　m はλ-計算の表式、ϕは線形論理の論理式

意味構成子によって、次の(6)のような基本的な文の意味表示が線形論理における証明の過程として(7)のように進行するわけである。

(6)　John respects Mary.

(7)
$$\dfrac{\dfrac{\mathbf{respect} : e1 \multimap (e2 \multimap t3) \qquad \mathbf{m} : e1}{\mathbf{respect(m)} : e2 \multimap t3} \qquad \mathbf{j} : e2}{\mathbf{respect(m)(j)} : t3}$$

(7)において意味構成子の右側の要素である線形論理の論理式を見ると、前節の(4)で見た線形論理版の Modus Ponens が 2 回適用されて、最終的に「結論」t3 が得られていることが分かる。

この形式論理版の Modus Ponens は、推論規則としては含意(implication)の除去であるが、逆に含意の導入が、対応する意味構成子の左側としてはλ-抽象(λ-abstraction)に相当することが分かる。(8)にこの対応を示す。

(8)

$$\frac{\mathbf{respect(y)(x)} : t3}{\lambda \mathbf{x.respect(y)(x)} : e2 \multimap t3} \qquad\qquad \frac{\mathbf{respect(y)(x)} : t3}{\lambda \mathbf{y.respect(y)(x)} : e1 \multimap t3}$$

この時、当然のことながら、含意の導入の操作としては(8)の左側の展開も右側の展開も同等に可能であることに注意したい。

このように、λ-計算側（意味構成子の左側）としてはλ-抽象に相当する操作が、線形論理側では含意の導入に相当し、線形論理の構成から見て、自由変項の数だけこの操作の可能性が存在する

この含意の導入の複数の可能性を活用することによって、量化表現を含んだ文における量化のスコープの問題についても、Glue 意味論においては極めて自然な処理が可能となるのである。このことを、次の(9)の文の Glue 意味論による処理を例として見てみよう。

(9) Someone respects everyone.

改めて述べるまでもなく、(9)には someone と everyone のスコープ関係について、表層の語順の通りのもの（someone が everyone に対してより広いスコープを取る）と表層の語順の逆のもの（everyone が someone に対してより広いスコープを取る。inverse scope interpretation と呼ばれることが多い。）との 2 通りの可能性が存在する。

このスコープ関係をどのように導出するかは、意味論にとって本質的とも言える重要性があり、意味論のそれぞれのフレームワークにおいて独自の分析の方法があるが、Glue 意味論においては、線形論理の証明の過程を反映して、次の(10)から(13)のように導出される。(10), (11)は Gothem(2018)の Figure 3、(12), (13)は Figure 4 を、それぞれ基本として表記したものである。

(10)

$$\underline{\mathbf{respect} : e1 \multimap (e2 \multimap t3) \quad [z : e1]}$$
$$\underline{\mathbf{respect(z)} : e2 \multimap t3 \quad [v : e2]}$$
$$\underline{\mathbf{respect(z)(v)} : t3}$$
$$\underline{\lambda \mathbf{P.\forall x.person(x) \rightarrow P(x)} : (e1 \multimap t3) \multimap t3 \qquad \lambda \mathbf{z.respect(z)(v)} : e1 \multimap t3}$$
$$\underline{\mathbf{\forall x.person(x) \rightarrow respect(x)(v)} : t3}$$
$$\lambda \mathbf{v.\forall x.person(x) \rightarrow respect(x)(v)} : e2 \multimap t3 \; = \; \alpha$$

(11)

$$\underline{\lambda \mathbf{Q.\exists y.person(y) \wedge Q(y)} : (e2 \multimap t3) \multimap t3 \qquad \alpha}$$
$$\mathbf{\exists y.person(y) \wedge (\forall x.person(x) \rightarrow respect(x)(y))} : t3$$

(12)

$$\underline{\mathbf{respect} : e1 \multimap (e2 \multimap t3) \quad [z : e1]}$$

$$\underline{\lambda Q.\exists y.\mathbf{person}(y)\wedge Q(y) : (e2 \multimap t3) \multimap t3 \qquad \mathbf{respect}(z) : e2 \multimap t3}$$
$$\underline{\exists y.\mathbf{person}(y)\wedge \mathbf{respect}(z)(y) : t3}$$
$$\lambda z.\exists y.\mathbf{person}(y)\wedge \mathbf{respect}(z)(y) : e1 \multimap t3 \ = \ \beta$$

(13)
$$\underline{\lambda P.\forall x.\mathbf{person}(x)\rightarrow P(x) : (e1 \multimap t3) \multimap t3 \qquad\qquad \beta}$$
$$\forall x.\mathbf{person}(x)\rightarrow(\exists y.\mathbf{person}(y)\wedge \mathbf{respect}(x)(y)) : t3$$

(10), (11)が表層語順の通りのスコープ関係の意味解釈、(12), (13)が表層語順とは逆のスコープ関係の意味解釈(inverse scope interpretation)をそれぞれ導出している。各意味構成子がすべてλ-表式と線形論理の論理式のペアとなっており、導出の途中の段階では1行に相当の横幅を要することから、いずれの導出も2段階に分割して表示した。

Glue 意味論によるスコープ関係の処理のもう一つの例として、次の(14)について見てみよう。

(14)　A fan of every band cheered.

よく知られているとおり、(14)には(15a, b)に示す2通りの異なる意味解釈がある。

(15)　a.　すべてのバンドを好きな一人のファンが喝采した。
　　　b.　どのバンドについても、そのバンドのファンが喝采した。

(15a)の意味解釈に至る Glue 意味論の導出過程は、(16), (17)のようになる。上の(10), (11)と(12), (13)の場合と同じく、導出過程を2段階に分割して表示している。さらに、(16)においては、スペースの関係上、導出過程の大筋には関わらない技術的な細部をいくつか省略して表記している。(16), (17)は Gotham(2018)の Figure 11、(18), (19)は Figure 12 を、それぞれ基本として表記したものである。

(16)
$$\underline{\mathbf{fan} : e2 \multimap (e3 \multimap t3) \qquad [u : e2]}$$
$$\underline{\mathbf{fan}(u) : e3 \multimap t3 \qquad [v : e3]}$$
$$\underline{\mathbf{fan}(u)(v) : t3}$$
$$\underline{\lambda G.\forall y.\mathbf{band}(y)\rightarrow G(y) : (e1 \multimap t3) \multimap t3\text{[2]} \qquad\qquad \lambda u.\mathbf{fan}(u)(v) : e1 \multimap t3}$$
$$\underline{\forall y.\mathbf{band}(y)\rightarrow \mathbf{fan}(y)(v) : t3}$$
$$\lambda v.\forall y.\mathbf{band}(y)\rightarrow \mathbf{fan}(y)(v) : e3 \multimap t3 \ = \ \alpha$$

---

[2] スペースの都合上、この表式に至る every と band との組み合わせの処理の詳細を省略している。

(17) $\underline{\lambda P.\lambda Q.\exists x.P(x)\wedge Q(x):(e3\multimap t3)\multimap((e3\multimap t3)\multimap t3) \qquad \alpha}$

$\underline{\lambda Q.\exists x.\forall y.(band(y)\rightarrow fan(y)\ (x))\wedge Q(x):(e3\multimap t3)\multimap t3 \quad cheered:e3\multimap t3}$

$\exists x.(\forall y.(band(y)\rightarrow fan(y)\ (x)))\wedge cheered(x):t3$

他方、(15b)の意味解釈に至る Glue 意味論の導出過程は、(18), (19)のようになる。

(18) $\underline{fan:e2\multimap(e3\multimap t3) \quad [u:e2]}$

$\underline{\lambda P.\lambda Q.\exists x.P(x)\wedge Q(x):(e3\multimap t3)\multimap((e3\multimap t3)\multimap t3) \qquad fan(u):e3\multimap t3}$

$\underline{\lambda Q.\exists x.fan(u)(x)\wedge Q(x):(e3\multimap t3)\multimap t3 \qquad [R:e3\multimap t3]}$

$\underline{\exists x.fan(u)(x)\wedge R(x):t3}$

$\lambda u.\exists x.fan(u)(x)\wedge R(x):e3\multimap t3 = \beta$

(19) $\underline{\lambda G.\forall y.band(y)\rightarrow G(y):(e3\multimap t3)\multimap t3 \qquad \beta}$

$\forall y.band(y)\rightarrow(\exists x.fan(y)(x)\wedge R(x)):t3$

$\underline{\lambda R.\forall y.band(y)\rightarrow(\exists x.fan(y)(x)\wedge R(x)):(e3\multimap t3)\multimap t3 \quad cheered:e3\multimap t3}$

$\forall y.band(y)\rightarrow(\exists x.fan(y)(x)\wedge cheered(x)):t3$

4. Glue 意味論の可能性

以上見てきたようにGlue意味論では意味解釈に至る過程が線形論理における証明という形を取り、このことはそもそも線形論理が厳密に構成的な論理として構想されたということを考え合わせると、現在のMinimalist統語論に対応する意味論として、Glue意味論は親和性が高いと言える。つまり、線形論理の証明のプロセスが厳密に構成的であることから、前節で見た意味解釈の導出でも明らかなとおり、導出に至る各ステップが言わば「ブロック」のように積み重なっている。現在のMinimalist統語論では、統語構造を作り上げる過程でphaseという言わば「単位(unit)」のような段階があり、統語構造の組み上げがphaseに達した段階で、それまでに構成された統語構造が意味論サイドに送られ、意味論サイドではこれを入力として意味解釈の計算(computation)を遂行することとなる。ところが、Montague以来の伝統的な形式意味論では、意味解釈の過程が、トップダウンに処理を進めるにせよボトムアップに処理を進めるにせよ、タイプtのレベル（真偽の判定が可能となる命題のレベルで、統語論では基本的にIPノードに相当する）が言わば「核」となって遂行される。とりわけ、トップダウンの処理では、タイプtの命題レベルからではなく途中の段階から処理を開始した場合、統語論サイドからより上位のphaseからの出力が意味論サイドに送られてきた時に、どのような形で意味解釈の計算が続行されるのか、必ずしも明らかではない。Heim and Kratzer(1998)などに代表される、生成文法のコンセプトに対応している形式意味論の標準的な教科書においても、意味解釈はタイプtまで完成された統

語構造を対象として行なわれることを前提としており、Minimalist統語論におけるphaseごとの処理に意味論サイドがどのように対応するのかについては触れられていない[3]。

これに対して、Glue 意味論では、統語論サイドから phase ごとに送られてくる統語構造を言わば「チャンク」として、各「チャンク」ごとに構成的に処理することができる。phase ごとに意味処理の「チャンク」ができることになるが、ここで線形論理の厳密構成性が大きな力を発揮し、各「チャンク」ごとの処理がタイプの照合を中心として自然な形で進行できることはもちろんのこと、処理が終了した「チャンク」ごとの関係も、線形論理の構成性によって適切な形で照合できることになる。この時に、繰り返しになるが線形論理の厳密構成性により、複数の「チャンク」を組み合わせる可能性について、線形論理の証明の過程を適切に進行させることができる組み合わせ以外では証明を完成させることができないことから、必然的に、可能な組み合わせは極めて限定されることとなり、その組み合わせのおのおのに対応して証明が進行して、最終的に意味解釈が導出されることとなる。こうしたことから、Glue 意味論が統語論サイドから phase ごとに入力（統語論サイドから見れば出力）を受け入れて意味解釈を進める意味論のモデルとして、優れた性質を備えていることが分かる。

また、筆者は近年の一連の論考（三藤(2013, 2015, 2017)）において、意味論における計算(computation)について、とりわけ生成文法の理論的枠組みの下での構想について議論してきた。その中で、三藤(2013)においては、現在の Minimalist 統語論では Merge の操作は binary であると規定されているが、少なくとも意味論サイドからの考察による限りはこのことを「経済性」の観点から正当化することは極めて困難であることを論じた。しかし、Glue 意味論がベースとしている線形論理がコントラクション(contraction)と弱化(weakening)を認めない体系となっており、証明の展開は第 2 節の(4)に示した線形論理版の Modus Ponens が基軸となって進められることから、第 3 節で具体例を見たとおり、照明の各ステップごとに用いられる前提は必然的に 1 つだけということになり、このことが意味論サイドから見ても Merge 操作が binary であることの裏付けとなる可能性が生じてくる。ただし、三藤(2013)における意味論サイドからの議論では、連言(conjunction)や選言(disjunction)について、それぞれの演算子（すなわち、$\wedge$と$\vee$）が $P \wedge Q$, $P \vee Q$ のように 2 つの命題 P と Q とを項に取ると見るのがこの演算の最も自然な捉え方であり、schönfinkelization を用いて binary 構造が 2 つ積み重なった形に「変形」するのは意味論サイドの自然さを犠牲にして Minimalist 統語論の Merge 操作で想定されている binary 構造に適合させるための言わば「便法」と言わざるを得ないことを指摘した。この観点から見ると、Glue 意味論において連言や選言をどのように取り扱うのが最も自然な形式となるかが大きな問題点となってくる。

---

[3] もちろん改めて言うまでもなく、Heim and Kratzer(1998)が執筆されていた時期には、生成文法統語論において phase の考え方は未だ明確に確立されてはいなかったのであるから、phase への対応に関して述べられていないのは当然のことと言える。

さらに、これまで見てきたとおり、Glue 意味論は意味解釈の過程を線形論理における証明の展開に対応させて構築していく点に最大の特徴があると言えるが、実際に人間が自然言語の意味解釈をどのような形で遂行しているのかという、意味解釈に関する「心理的実在性(psychological reality)」の観点、また筆者が近年考察してきた「モデル（近年の科学哲学において重要視されている）としての意味論」という観点（三藤(2017)を参照）、この二つのいずれの観点から見ても、意味解釈の過程を論理における証明の過程と同一視するという Glue 意味論の立場は、Montague 以来の伝統的な形式意味論に比べて優れていると論じることは難しいと言わざるを得ない。

このような問題点は含みつつも、Minimalist 統語論に対する意味論としての Glue 意味論は Gotham(2018)によってその緒に就いたばかりであると言っても過言ではないことから、筆者も引き続き Glue 意味論の可能性について考察を進めていきたいと考えている。

5. おわりに

以上見てきたように本稿においては、線形論理をベースとする Glue 意味論を Minimalist 統語論から入力を受け取る意味論として構築する試みを Gotham(2018)に基づいて概観し、線形論理の厳密な構成性により、Minimalist 統語論で想定されている phase ごとの意味解釈の構成に非常に適した特性を有することを指摘した上で、Glue 意味論の更なる可能性について、また簡潔ではあるが問題点についても考察した。

前節の最後にも述べたが、筆者も今後 Glue 意味論の可能性について考察を深めていきたいと考えている。

参考文献

三藤　博(2007)「イベント意味論に基づく日本語意味論の構築に向けて」『自然言語への理論的アプローチ』（大阪大学言語文化共同プロジェクト2006）　61-70.

三藤　博(2013)「意味論の基礎についての一考察」『自然言語への理論的アプローチ』（大阪大学言語文化共同プロジェクト2012）　41-48.

三藤　博(2015)「「理論の意味論的捉え方」と言語学」『自然言語への理論的アプローチ』（大阪大学言語文化共同プロジェクト2014）　41-48.

三藤　博(2017)「意味論におけるモデル構成を再考する—Nefdt(2016)を受けて—」『自然言語への理論的アプローチ』（大阪大学言語文化共同プロジェクト2016）　59-67.

Gotham, Mathew (2018) Making logical form type-logical : Glue semantics for minimalist syntax. *Linguistics and Philosophy* 41: 511-556.

Heim, Irene and Angelika Kratzer(1998) *Semantics in generative grammar.* Oxford: Blackwell.