



ことばと意味の「構成性」について

カテゴリー論と意味の形式的理論

はじめに

ことばと意味の「構成性」について



カテゴリー論と意味の形式的理論

「ことばと意味の構成性」

Compositionality

今回のマルレクは、ことばと意味の関係を「構成性」という観点から取り上げたいと思います。

「構成性」、あまり聞きなれない言葉かもしれませんが。英語だと、"*compositionality*"と言います。あるものが、もっと基本的なものから「構成」されていることを言います。

「構成性」の例

例えば、 $1 + 2 = 3$ という式があったとします。この時、この式は、"1", "2", "3" という数字と、 "+", "=" という記号から「構成」されていると考えることができます。

コンピュータのプログラムでも同じです。 $x = x + 1$ という代入文は、変数を表す "x" と、数字を表す "1" と、 "+", "=" という記号から「構成」されたものです。どんな長いプログラムも、それが、構成されたものであることは変わりありません。

ことばの「構成性」

こうした「構成性」は、いろんなところで見ることができます。

ことばも、そうした「構成性」を持っています。

例えば、「黒い猫が走る」ということばは、「黒い」という形容詞、「猫」という名詞、「が」という助詞、「走る」という動詞から構成されています。

文法という「構成ルール」

ここまでで挙げた「構成」の例は、それぞれ違う「構成ルール」に基づくものですが、少し乱暴ですが、それらの「構成ルール」を、「数式の文法」、「プログラムの文法」、「ことばの文法」というように、「文法」と呼ぶことができます。

(別の言い方もあります。こうしたレベルの記号や語からの「構成ルール」を「統語規則」とか「syntax」ということがあります。)

意味の「構成性」

それでは、「意味」にも、「構成性」があるのでしょうか？

ことばを例に考えれば、ことばは意味を運ぶものです。その意味では、ことばと意味との関係は、直接的で、ことばと意味を分けるのは、少し難しいかもしれません。

ただ、「黒い猫」の意味は、「黒い」という語の意味と「猫」という語の意味から「構成」されると考えることは自然なことのように思えます。

「（黒い猫）（が）（走る）」という文の意味も、（黒い猫）と（走る）の意味から「構成」されると考えるのは可能です。

意味は、どう表現されるか？

意味の「構成性」を考える時難しいのは、構成された「意味」は、もはやもとの記号や語の並びそのものとは違うものだと言うことはわかっているのですが、それを、うまく表現することが難しいことかもしれません。

先にも述べたように、語のレベルで言うと特にそうなのですが、語そのものと語の意味は、強く結びついていて、分離するのは少し面倒です。でも、語から構成された文になると、文そのものとその意味は、分離しやすくなります。

ことばと意味の「構成性 (compositionality)」 基本的な観察

- 文は文法に従って語から構成される
- 文の意味は、語の意味に依存する
 - 文法的に正しいが意味がない文もある
Colorless green ideas sleep furiously.
- 文の意味は、文の文法的構成に依存する
 - 二つの意味を持つ、あいまいな文
I saw a man with a telescope.

多様な「自然な意味」

記号や語の構成ルールに従った構成、文法的なレベルでの構成物は、そのうえ、コンテキストに応じて様々な「自然な意味」を持ちます。

プログラムでの、 $x = x + 1$ という代入文は、「 x という変数に $x + 1$ という値を代入する」という意味を持つと考えるのが自然です。

また、 $1 + 2 = 3$ という数式の意味は、「 $1 + 2 = 3$ 」は正しい計算である」と考えることができます。

「意味」を一つの枠組みで捉える

ただ、これらの例でも、「意味」の「解釈」は、最初の直接的対象であった、「文法的構成」のそれぞれの要素あるいはその全体を、「意味」の構成に取り込んでいるように見えます。それぞれの例でも、意味の「構成性」は、透けて見えています。

これらの、いくつかの「意味」の「解釈」を、一つの枠組みで捉えることができるでしょうか？

ことばと意味の「構成性」について

Agenda

はじめに

第一部 意味の理論入門

第二部 ことばの構成性
-- 文法 --

第三部 意味の構成性
-- カテゴリー論的分散意味論 --

ことばと意味の「構成性」について

第一部 意味の理論入門

- 「ルール」の世界と「たとえ」の世界
- 「理論」と「モデル」
- Functorial Semantics

ことばと意味の「構成性」について

第二部 ことばの構成性

-- 文法 --

- Minimalist Program
- ChomskyとLambek
- Combinatory Categorical Grammar
- Pregroup Grammar

ことばと意味の「構成性」について

第三部 意味の構成性

-- カテゴリー論的分散意味論 --

- カテゴリー論的構成的分散意味論 Overview
- DisCoCat入門
- DisCoCat -- Coecke's original form
- Interlude -- Monoidal categoryとString Diagram
- DisCoCat -- Coecke's diagrammatic calculus



第一部

意味の理論入門

カテゴリー論と意味の形式的理論

ことばと意味の「構成性」について

第一部 意味の理論入門

- 「ルール」の世界と「たとえ」の世界
- 「理論」と「モデル」
- Functorial Semantics

「ルール」の世界と「たとえ」の世界



カテゴリー論と意味の形式的理論

多様な「意味」の多様な「解釈」を、 一つの枠組みで捉えることができるか？

前回は、こう言う問いで終わったのだが、いくつかの準備なしでは、それにすぐに答えるのは難しい。

このセミナーの目的の一つは、ことばの構成性と意味の構成性を理論的に結びつけることなのだが、今回のセッションでは、すこし、回り道をしようと思う。

意味を理解するために、我々が行うこと 小学校の算数の場合

まず、ある表現を、その表現とは異なる形で、その意味を解釈し理解しようとする我々の認識の働きに注目しようと思う。

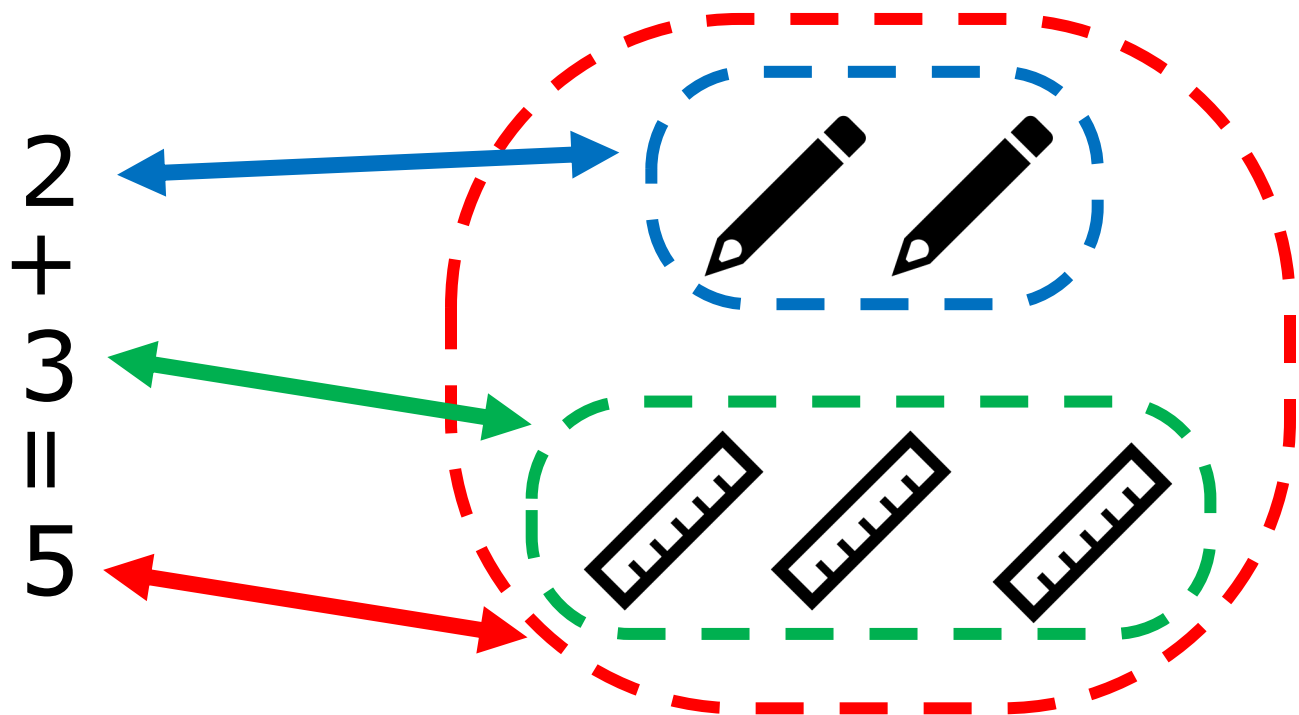
それは、素朴なものではあるが、与えられた表現から、その表現が指し示す「意味」を取り出そうとするでプリミティブな試みとみなすことができる。

我々は、まだ、ことばの構成ルールを記述していない。

ここでは、その構成ルールも意味も、比較的明確な、小学生が習う算数を例に考えてみようと思う。

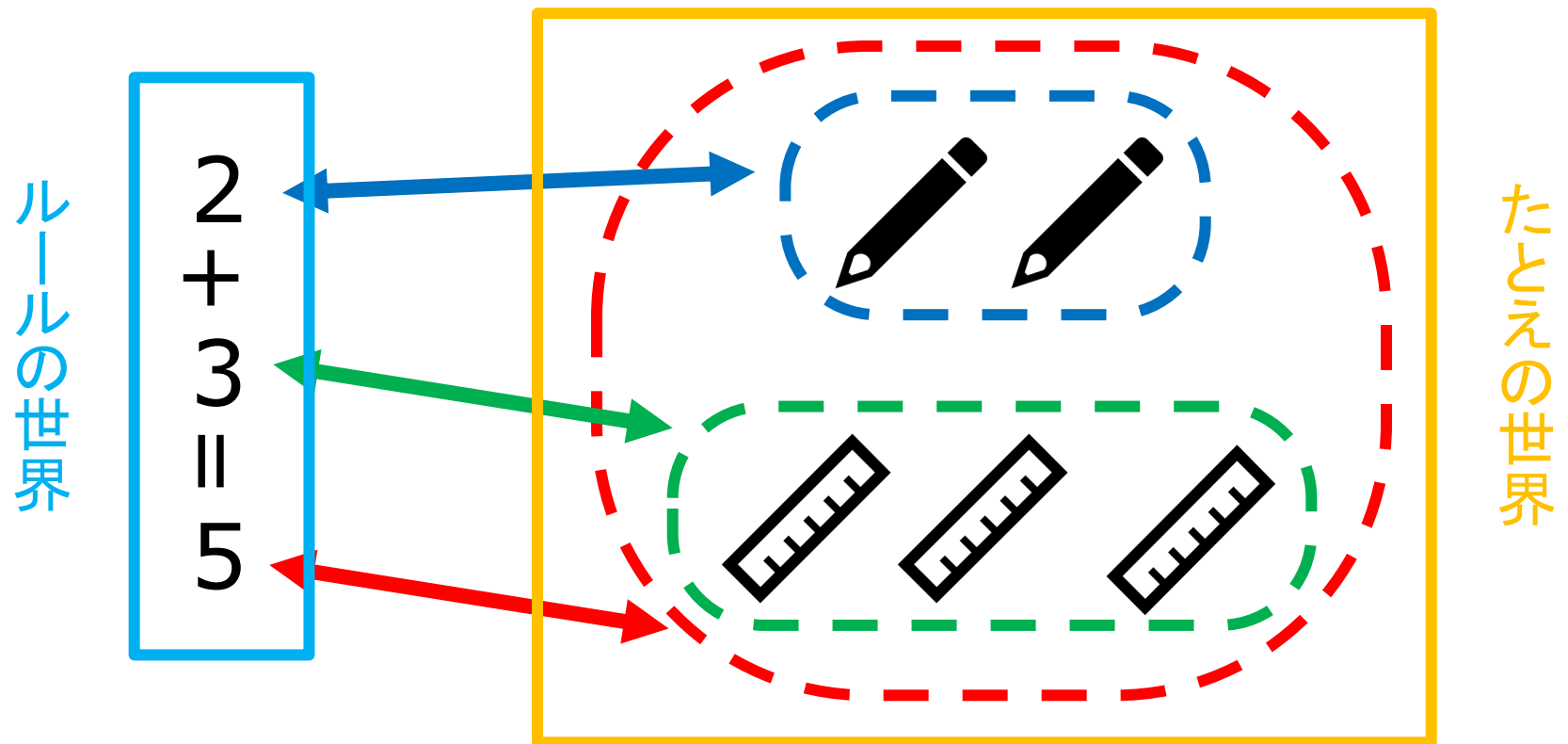
2 + 3 = 5 の計算の理解

2+3=5となるのを、「鉛筆二本と、定規三つで、全部で五つ」というように、具体的なものをあげて、「たとえ」で説明することがある。



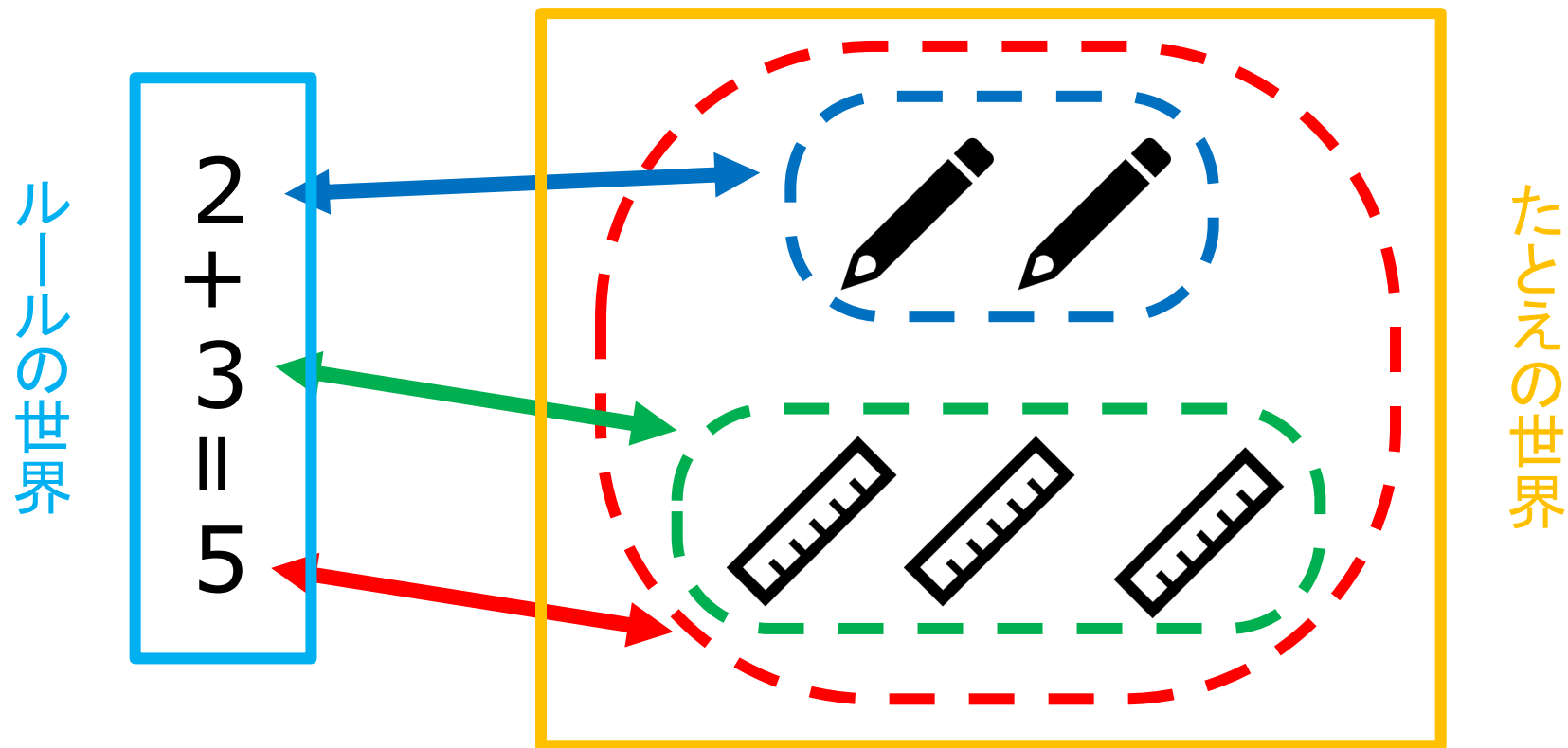
「ルールの世界」と「たとえの世界」

左側を「ルールの世界」、右側を「たとえの世界」と呼ぶことにしよう。



「ルールの世界」と「たとえの世界」の関係

この二つの世界は、お互いに関連している。「ルールの世界」は「たとえの世界」に、具体的対応物を持ち、「たとえの世界」は「ルールの世界」に抽象的な対応物を持つ。



「たとえ」で、「ルール」の「意味」を理解する

数学のルールは、基本的には現実の具体的な関係から抽象されたものである。理解しやすい具体的関係から、ルールを導入・説明するのは有効である。

「ルール」は、抽象的なものだが、「たとえ」は、ルールの「意味」を理解させる。

先の「鉛筆二本と、定規三つで、全部で五つ」という例では、実は、「全部で」という言葉が、足し算の説明になっている。

「たとえ」では、具体的なものにたいする理解とともに、「言葉」の理解が、前提になることがある。

「ルール」で理解する vs. 「たとえ」で理解する

「たとえ」の世界は、具体的で現実的である。その世界は、我々の直観と「言葉」によって理解可能な「意味」の世界によって支えられている。

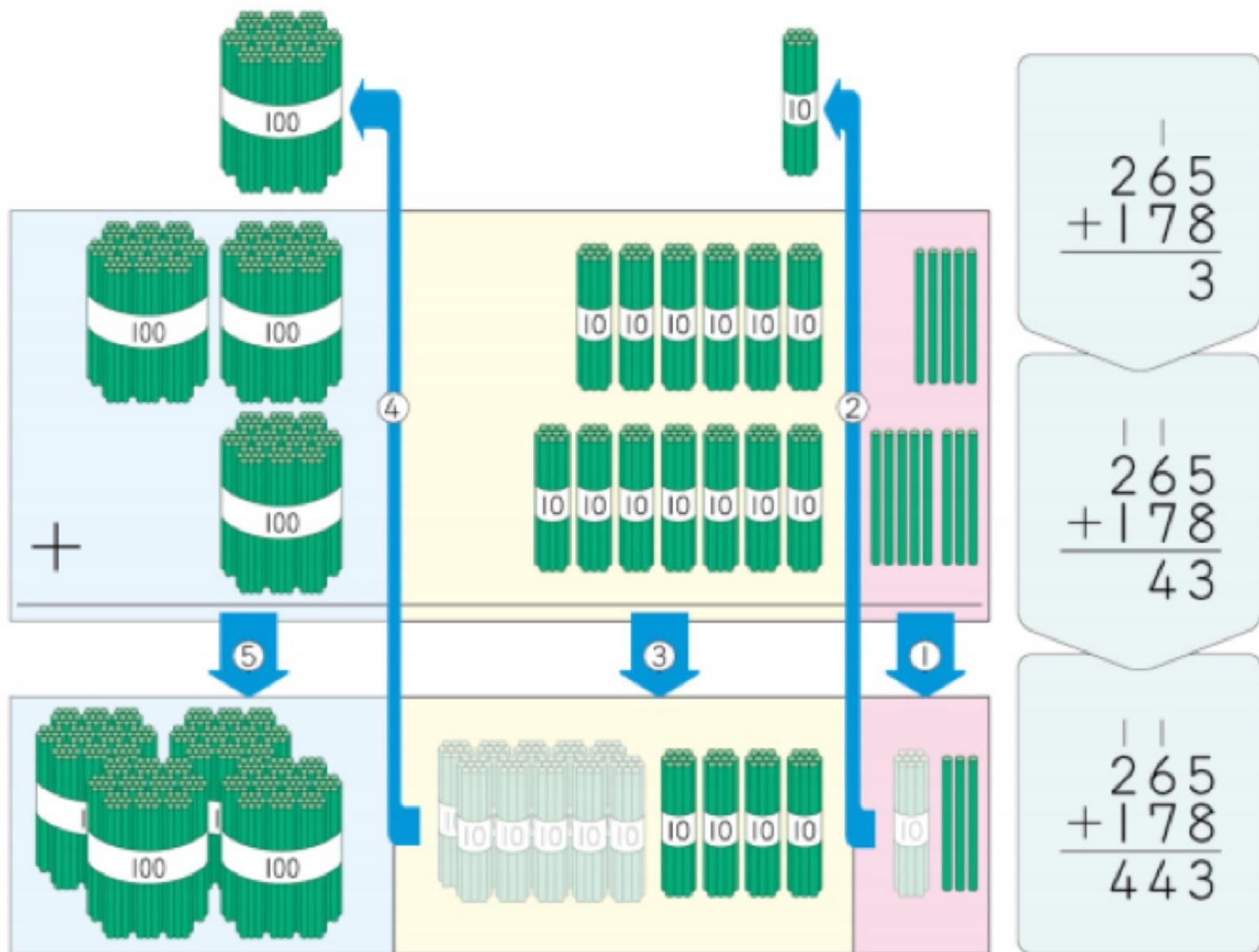
ただ、それは、複数の意味を持つかもしれないし、ひとつの「たとえ」で全ての「ルール」を説明できないという意味では個別的である。

また「たとえ」にはいろいろのタイプがある。言葉だけでなく、図解することも、「たとえ」による説明の一種である。言葉による「たとえ」と図形による「たとえ」では、抽象度が異なる。

3

265+178の筆算のしかたを考えましょう。

啓林館 { 3年上 p.57 }



「ルール」の世界の特徴

数学の計算では「ルール」を「アルゴリズム」という。その「ルール」に従って、きちんと計算すると、余計なことは考えなくても正しい計算結果が保証されるという限りでは、「ルール」の世界は、他の世界とは独立しているように見える。

ただ、「ルール」は、計算をするという目的のためには万能かもしれない。ただ、計算「ルール」というものは、具体的なようで我々の頭の中にしかない抽象的なものであって、それ自体では意味を持たないようにも見える。

数学は、基本的には、それ自身の原理に基づいた自律的な「ルール」の世界である。(実際には、それは単純化しすぎであるのだが。)

4

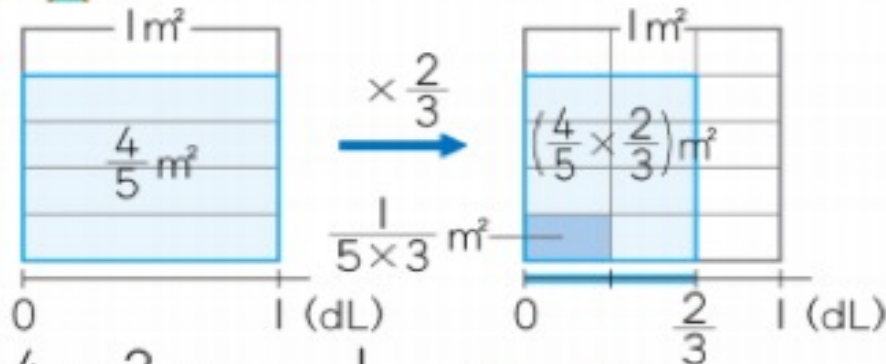
1dLで $\frac{4}{5}\text{m}^2$ ぬれるペンキがあります。
 $\frac{2}{3}\text{dL}$ のペンキでは何 m^2 ぬれますか。




6年 p.280



ひなた



$\frac{4}{5} \times \frac{2}{3}$ は、 $\frac{1}{5 \times 3}$ の  が

(4×2)個分だから、

$$\frac{4}{5} \times \frac{2}{3} = \frac{\square \times \square}{\square \times \square}$$

$$= \frac{\square}{\square} \text{m}^2$$

だいち



$$\begin{aligned} \frac{4}{5} \times \frac{2}{3} &= \frac{4}{5} \times \left(\frac{1}{3} \times 2\right) \\ &= \left(\frac{4}{5} \times \frac{1}{3}\right) \times 2 \\ &= \frac{4}{5 \times 3} \times 2 \end{aligned}$$

$$= \frac{\square \times \square}{\square \times \square}$$

$$= \frac{\square}{\square}$$

$$\frac{\square}{\square} \text{m}^2$$

分数の掛け算のたとえ $\frac{4}{5} \times \frac{2}{3} = \frac{8}{15}$ の場合

分数の割り算のたとえ $\frac{1}{2} \div \frac{2}{3} = \frac{3}{4}$ の場合

鉛筆が12本入ったケースと、3人で1組のグループがあったとする。この時、ケースの $\frac{1}{2}$ をグループの $\frac{2}{3}$ で分けたとしよう。

1グループあたりのケースを求めよ、

鉛筆6本($12 \times \frac{1}{2}$)を2人($3 \times \frac{2}{3}$)で分けたことになるので、グループの一人は3本($6/2$)の鉛筆を取ったことになる。この取り方で、一つのグループあたり何本の鉛筆が分配されたかを考えれば、グループは、3人で1組なので、9本(3×3)の鉛筆をえたことになる。これは、ケースで考えれば、 $9/12 = 3/4$ ケースになる。

$$\frac{1}{2} \div \frac{2}{3} = \frac{6}{2} \times \frac{3}{12} = \frac{3}{2 \times 2} = \frac{1 \times 3}{2 \times 2} = \frac{3}{4}$$

「理論」と「モデル」



カテゴリー論と意味の形式的理論

「理論」と「モデル」

ここでは、「ルールの世界」と「たとえの世界」の数学的形式化を紹介したい。それが、「理論」と「モデル」である。

ある世界で、基本的な公理と一階の述語論理の演繹規則が与えられているとする。この世界を「理論 Theory」と呼ぶことにしよう。この「理論」の世界で、ある命題 ϕ が成り立つのは、この理論の中で、命題 ϕ が証明される場合に限る。

今度は、もう一つ別の世界を考えよう。

この世界では、ある命題 ϕ が妥当する（「真」である）ことが、何らかの方法で定義されているとする。こうした世界を「モデル Model」と呼ぼう。

「理論」と「モデル」

ある理論TとあるモデルMについて、すべての命題 ϕ について、次の関係が成り立つとする。

Tにおいて ϕ は証明可能 \rightarrow Mにおいて ϕ は真である

この時、TはモデルMを持つ。あるいは、MはTのモデルである。という。

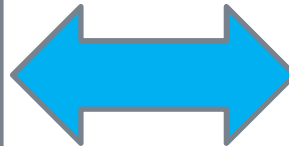
「理論」の世界と「モデル」の世界

理論 T

モデル M

形式的なルールが
与えられている。
その下で

命題 ϕ は証明可能



なんらかのルールが
この世界では成り立っていて
その下で

命題 ϕ は妥当する

「理論」と「モデル」

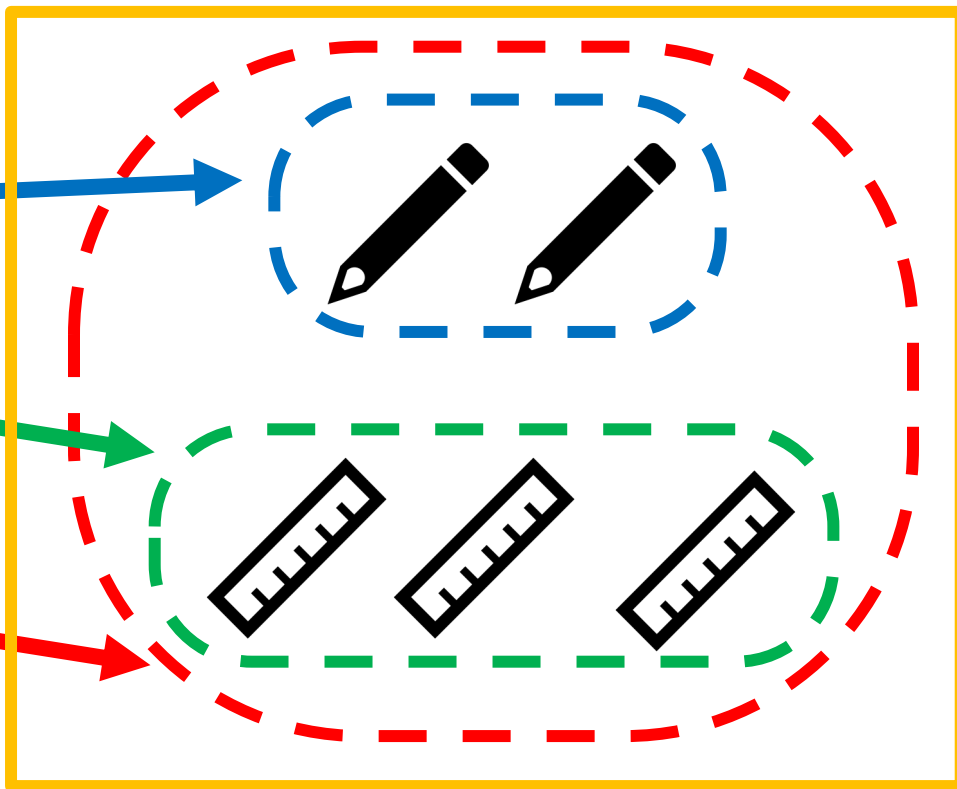
「ルールの世界」と「たとえの世界」の形式化

理論 T

モデル M

ルールの世界

$$\begin{array}{c} 2 \\ + \\ 3 \\ \hline 5 \end{array}$$

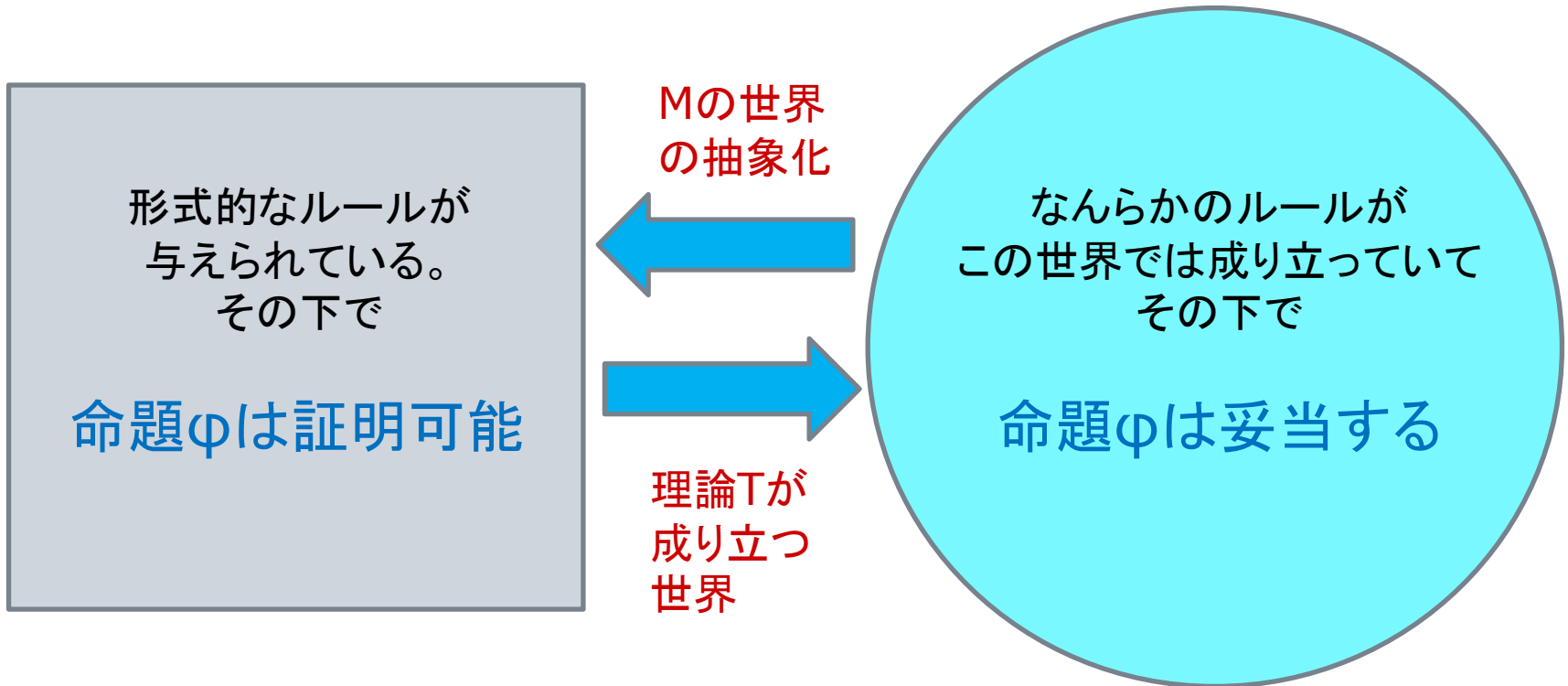


たとえの世界

「理論」の世界と「モデル」の世界

理論 T

モデル M

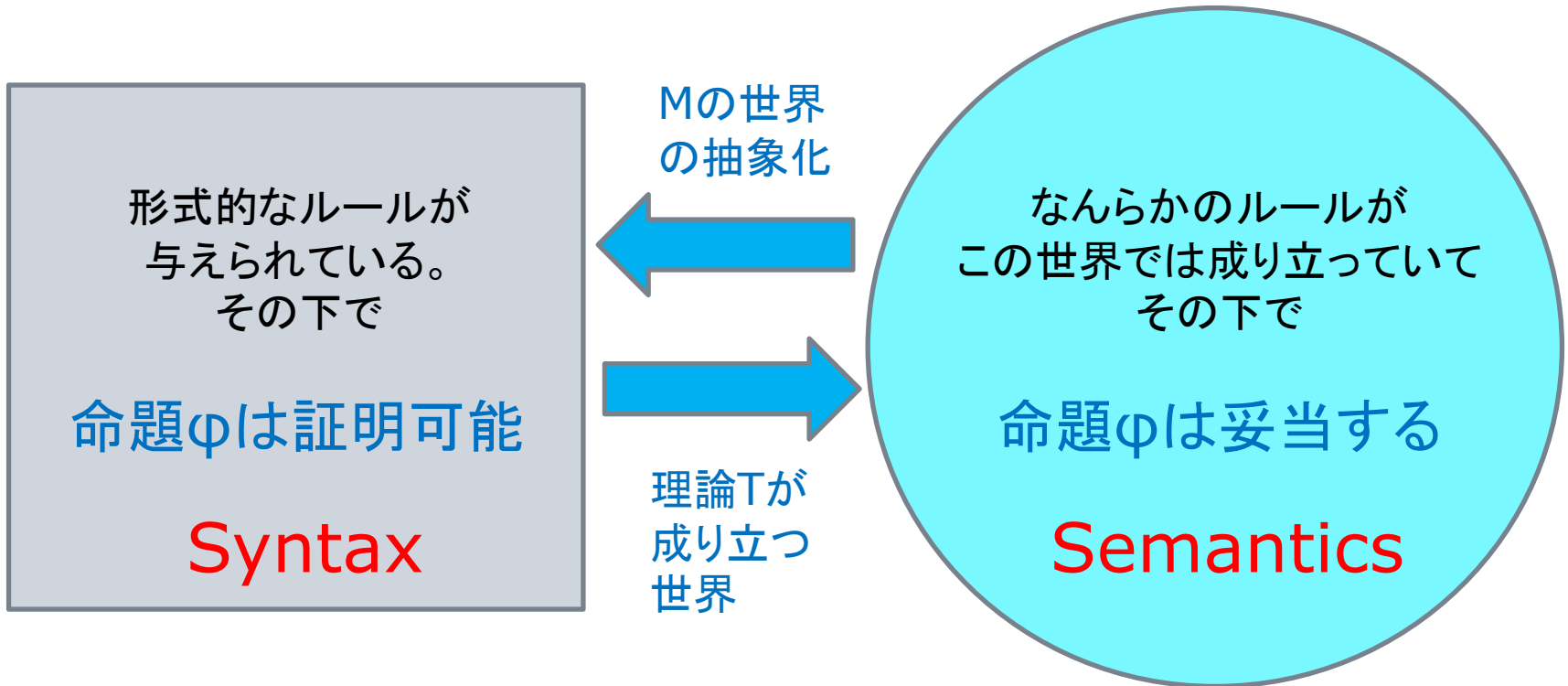


理論とモデルの「二重化」あるいは「双対性」

「理論」の世界と「モデル」の世界

理論 T

モデル M



モデルMは、理論Tに「意味」を与える

ゲーデルの完全性定理

理論とモデルの関係を、「すべてのモデルで真になる命題は、一階の述語論理で証明可能である」という形で初めて述べたのは、ゲーデルである。これをゲーデルの「完全性定理」という。

この定理から、「ある理論が無矛盾ならそれはモデルを持つ」ことが導かれる。<https://goo.gl/5HLMsK>

モデルの存在定理では、次の「レーベンハイム＝スコレームの定理」が有名である。

「理論が非可算濃度のモデルを持つなら、それは可算モデルも持つ。」

レーベンハイム・スコレムの定理 「モデルをもつなら、可算モデルも持つ」

Skolem's paradox is that

“every countable axiomatisation of set theory in first-order logic, if it is consistent, has a model that is countable.”

This appears contradictory because it is possible to prove, from those same axioms, a sentence that intuitively says (or that precisely says in the standard model of the theory) that there exist sets that are not countable. Thus the seeming contradiction is that a model that is itself countable, and which therefore contains only countable sets, satisfies the first order sentence that intuitively states "there are uncountable sets".

数学でのモデルの利用

MaruLabo「数理哲学への招待」

https://drive.google.com/file/d/1JJbm0MpF3os6yd2YP1Bv_HSAiOr37fg9/view?usp=sharing

非ユークリッド幾何学の発見

Lobachevskyは平行な仮定を否定することによって非ユークリッド幾何学を作成したが、Bolyaiはユークリッド幾何学と双曲幾何学の両方が、パラメータ k に依存するような幾何学を作り出した。

Bolyaiは物理宇宙の幾何学的構造がユークリッド的または非ユークリッド的であるとしても、それを数学的推論だけで決めることは不可能であるとして、それは物理学の課題だと述べて、彼の研究を終える。

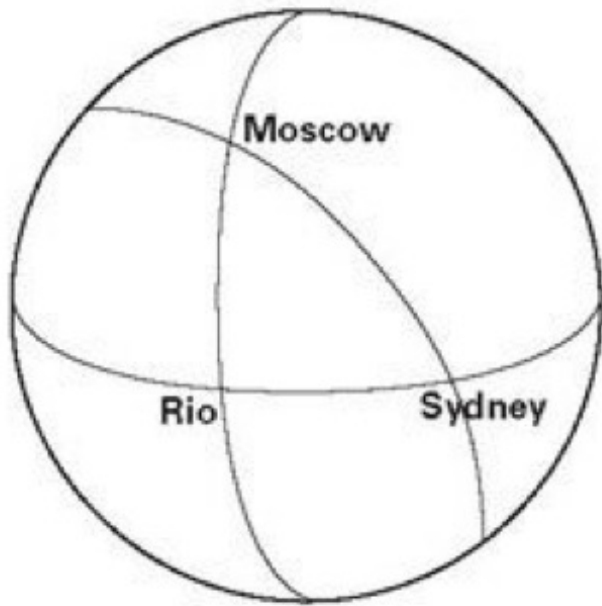
https://en.wikipedia.org/wiki/Non-Euclidean_geometry

リーマン

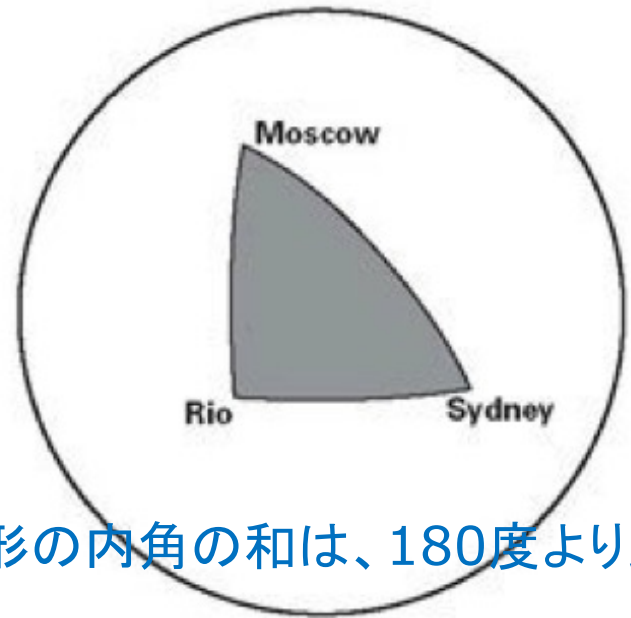
「幾何学の基礎をなす仮説について」

「周知のように幾何学は、空間の概念も、空間のなかでの幾何学的構成のための最初の基礎概念をも、何か与えられたものとして前提しています。幾何学はそれらに名目的定義を与えているだけであって、本質的な諸規定は公理の形式で現れています。だがその際、それら前提の関係は不明のままであって、それらの結合がはたして必然的であるのか、そしてどの程度まで必然的なのか、わからないし、またそれが可能であるかどうか、アプリアリにはわかりません」

幾何学が扱う事実は、「すべての事実と同じく、必然的ではなく、経験的に確実であるにすぎず、それらは仮説」である



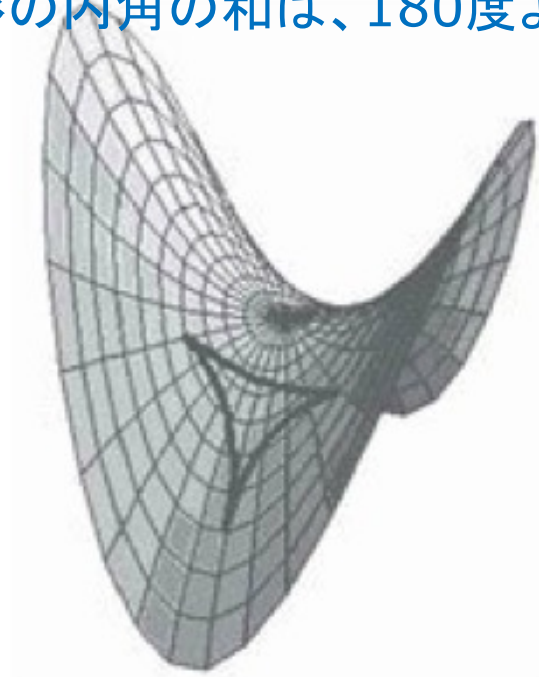
楕円幾何学



三角形の内角の和は、180度より大きい

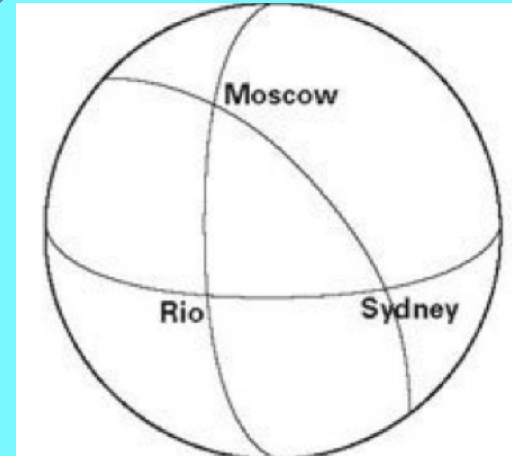
三角形の内角の和は、180度より小さい

双曲幾何学



リーマン幾何学

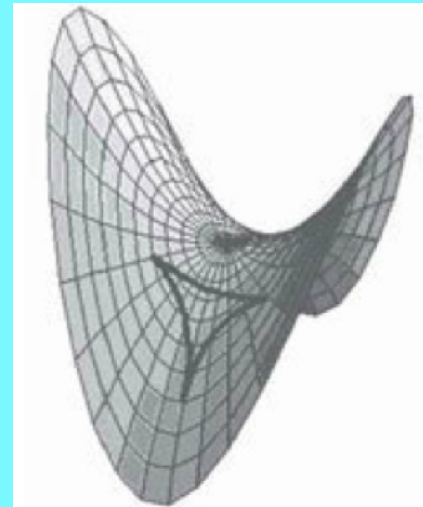
三角形の内角の和が、180度より大きい幾何学



理論

モデル

三角形の内角の和が、180度より小さい幾何学



アインシュタインの相対性理論へ

Flat spaces

Euclidean geometry \rightarrow Minkowski geometry

Kronecker δ tensor \rightarrow η tensor

Newton physics \rightarrow special relativity

Non flat spaces (always locally flat)

Curved metric \rightarrow gravitational field

Riemannian geometry \rightarrow Einstein general relativity

THE CONSISTENCY OF THE CONTINUUM HYPOTHESIS

KURT GÖDEL

²⁰ Gardner, R. S., A. J. Wahba, C. Basilio, R. S. Miller, P. Lengyel, and J. F. Speyer, these PROCEEDINGS, **48**, 2087 (1962).

²¹ Ralph, R. K., and H. G. Khorana, *J. Am. Chem. Soc.*, **83**, 2926 (1961).

²² Ames, B. N., and D. T. Dubin, *J. Biol. Chem.*, **235**, 769 (1960).

²³ Chen, P. S., Jr., T. Y. Toribara, and H. Warner, *Anal. Chem.*, **28**, 1756 (1956).

²⁴ Heppel, L. A., D. R. Harkness, and R. J. Hilmeo, *J. Biol. Chem.*, **237**, 841 (1962).

²⁵ Russell, W. E., and H. G. Khorana, *J. Biol. Chem.*, **234**, 2114 (1959).

²⁶ Kuy, E. R. M., N. S. Simmons, and A. L. Doucet, *J. Am. Chem. Soc.*, **74**, 1724 (1952).

²⁷ Sherman, J. R., and J. Adler, *J. Biol. Chem.*, **238**, 873 (1962).

²⁸ Waley, S. G., and J. Watson, *Biochem. J.*, **55**, 328 (1953).

²⁹ Akabori, S., K. Ohno, and K. Narita, *Bull. Chem. Soc. Japan*, **25**, 214 (1952).

³⁰ Lowry, O. H., N. J. Rosebrough, A. L. Farr, and R. J. Randall, *J. Biol. Chem.*, **193**, 265 (1951).

³¹ Nakamoto, T., and S. B. Weiss, these PROCEEDINGS, **48**, 880 (1962).

³² Fox, C. F., W. S. Robinson, and S. B. Weiss, *Fed. Proc.*, **22**, 463 (1963).

³³ Krakow, J. S., and S. Oshon, these PROCEEDINGS, **49**, 88 (1963).

THE INDEPENDENCE OF THE CONTINUUM HYPOTHESIS

By PAUL J. COHEN*

DEPARTMENT OF MATHEMATICS, STANFORD UNIVERSITY

Communicated by Kurt Gödel, September 30, 1963

This is the first of two notes in which we outline a proof of the fact that the Continuum Hypothesis cannot be derived from the other axioms of set theory, including the Axiom of Choice. Since Gödel¹ has shown that the Continuum Hypothesis is consistent with these axioms, the independence of the hypothesis is thus established. We shall work with the usual axioms for Zermelo-Fraenkel set theory,² and by Z-F we shall denote these axioms without the Axiom of Choice, (but with the Axiom of Regularity). By a model for Z-F we shall always mean a collection of actual sets with the usual ϵ -relation satisfying Z-F. We use the standard definitions³ for the set of integers ω , ordinal, and cardinal numbers.

非カントールの集合論の発見

Before giving details, we sketch the intuitive ideas involved. The starting point is the realization^{4, 5} that no formula $\alpha(x)$ can be shown from the axioms of Z-F to have the property that the collection of all x satisfying it form a model for Z-F in which the Axiom of Constructibility ($V = L$,⁶) fails. Thus, to find such models, it seems natural to strengthen Z-F by postulating the existence of a set which is a

ゲーデル

連続体仮説と集合論の無矛盾性の証明

「世紀の難問」と呼ばれた連続体仮説の問題で、大きな前進があったのは、1940年のことだ。クルト・ゲーデルが、連続体仮説と選択公理の二つを満たす集合論のモデルを実際に構成して見せたのだ。この集合論のモデルの中では、連続体仮説は「真」である。

この結果は、連続体仮説を直接に証明したものではないことに注意しよう。それは、集合論と連続体仮説には矛盾がないこと、すなわち、集合論の中では連続体仮説の否定は証明できないことを意味している。

コーエン 1963年 連続体仮説と集合論の独立性の証明

連続体仮説を最終的に解決したのは、J.P.コーエンだった。彼は、「強制法(Forcing Method)」というモデル構成法を創出し、**連続体仮説の否定が成り立つモデルを構成した。**

先のゲーデルの結果と合わせると、連続体仮説は、集合論の公理群からは独立であることが証明されたことになる。連続体仮説は、集合論の公理群からは証明できないのだ。

結果から見ると、カントールが、全力で連続体仮説を証明しようとして、成功しなかったのには理由があったのである。

ゲーデル
連続体仮説(GCH)
と集合論(ZF)の
無矛盾性証明



連続体仮説が
成り立つ集合論の
モデルの構築

コーエン
連続体仮説(GCH)
の集合論(ZF)からの
独立性証明



連続体仮説の否定が
成り立つ集合論の
モデルの構築

19世紀、20世紀の数学史上の
大発見のいくつかは、
理論内部の演繹の結果としてではなく、
理論に対してそのモデルを考える
ことで見出されていることに注目。

それは、理論のモデルを通じて、
理論の意味するものを考えるということが、
理論の大きな飛躍を生み出すということを示している。

それはまた、理論のモデルと目された
実在的対象としての自然に対する
認識の飛躍をもたらした。

ただ、それはまだ、数学や物理学の
内部の出来事であった。

理論とモデルの関係が、
意味の理論の基礎であるという認識は、
必ずしも、明確だったわけではない。

Functorial Semantics



カテゴリー論と意味の形式的理論

理論とモデルの関係を、数学的にとらえなおす Functorial Semantics

このセッションでは、理論とモデルの関係を、カテゴリー論的にとらえなおしたLawvereのFunctorial Semanticsを紹介する。

このFunctorial Semantics は、ことばとその意味に、数学的モデルを与えようとする今回のセミナーの基本的なツールとなる。

CategoryとFunctor

Categoryとは？

Categoryは、Object(対象)と、Object間の関係を示すArrow(射:mappingともいう)からなる。

対象 A, B, C, \dots を集合として、射 f, g, h, \dots を、集合の間に定義された関数としてイメージしてみるといい。Categoryは、それを抽象化したものである。

対象Aから 対象Bへの 射 f を、

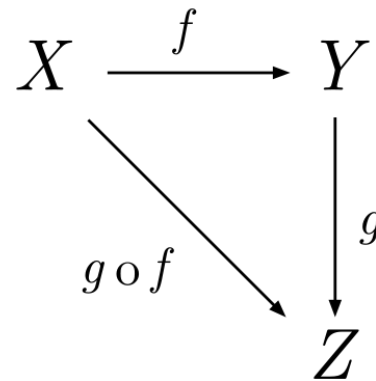
$$f: A \rightarrow B$$

とあらわす。

Categoryの満たすべき条件

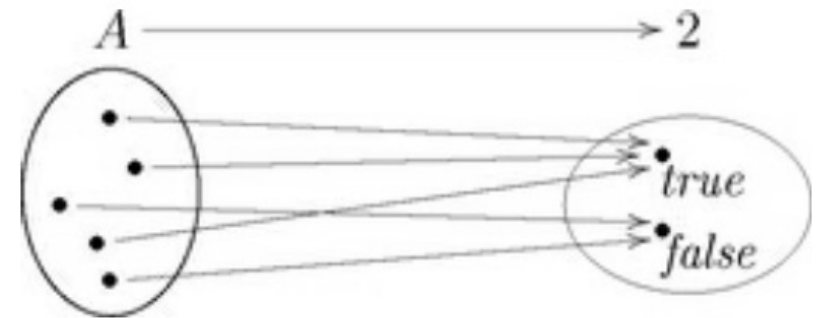
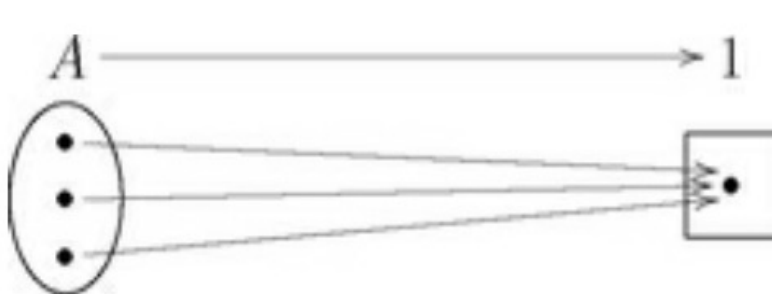
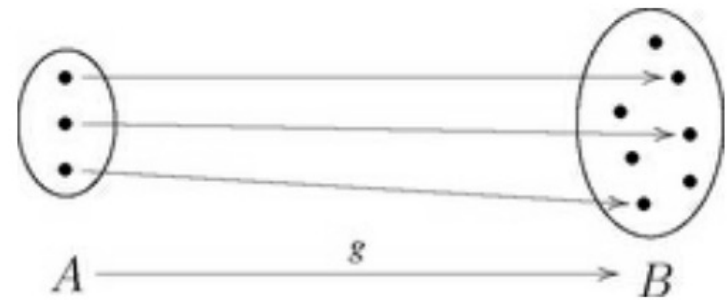
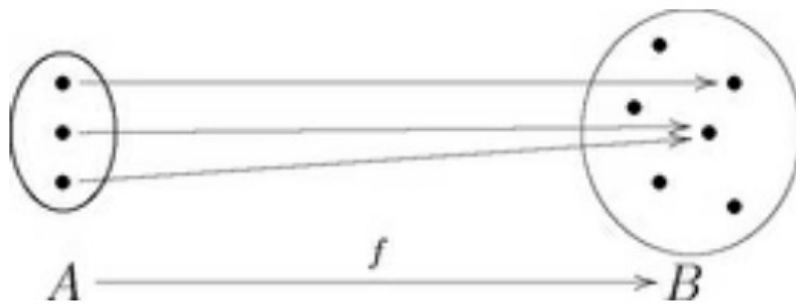
対象と射の集まりが、すべてCategoryになるわけではない。
Categoryは、次の条件を満たさねばならない。

1. $f: X \rightarrow Y, g: Y \rightarrow Z$ に対して、射の「合成」
 $g \circ f: X \rightarrow Z$ が存在する。

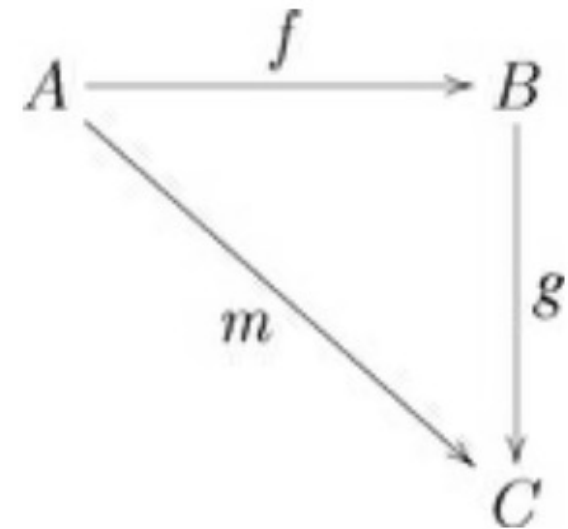
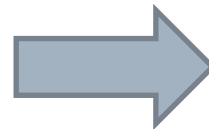
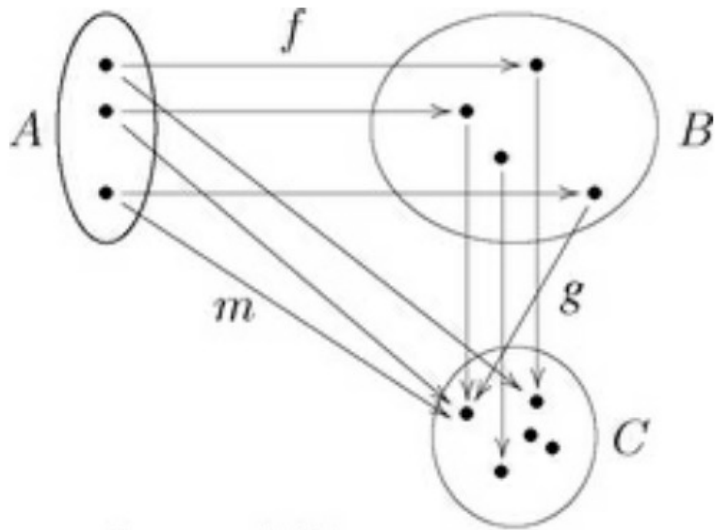


2. 全ての対象 X に対して、恒等射 $1_X: X \rightarrow X$ が存在して、
 $f: X \rightarrow Y$ に対して、 $f \circ 1_X = f = 1_Y \circ f$

対象(の内部)と射 $f: A \rightarrow B$ のイメージ



三つの対象 A, B, C と三つの射 f, g, m の例



Functor(関手)とは？

Category C と Category Dがあった時

- ・ Cの対象 X を Dの対象 $F(X)$ に
 - ・ Cの射 f を Dの射 $F(f)$ に
- 対応づける F をFunctorと呼ぶ。

Cの射 $f: X \rightarrow Y$ を Dの射 $F(f): F(X) \rightarrow F(Y)$ に対応させるものを、Covariant Functor(共変関手)と呼ぶ。

Cの射 $f: X \rightarrow Y$ を Dの射 $F(f): F(Y) \rightarrow F(X)$ に対応させるものを、Contravariant Functor(反変関手)と呼ぶ。

FUNCTORIAL SEMANTICS OF ALGEBRAIC THEORIES

F. WILLIAM LAWVERE

1963年

<https://www.pnas.org/content/pnas/50/5/869.full.pdf>

LawvereのFunctorial Semantics

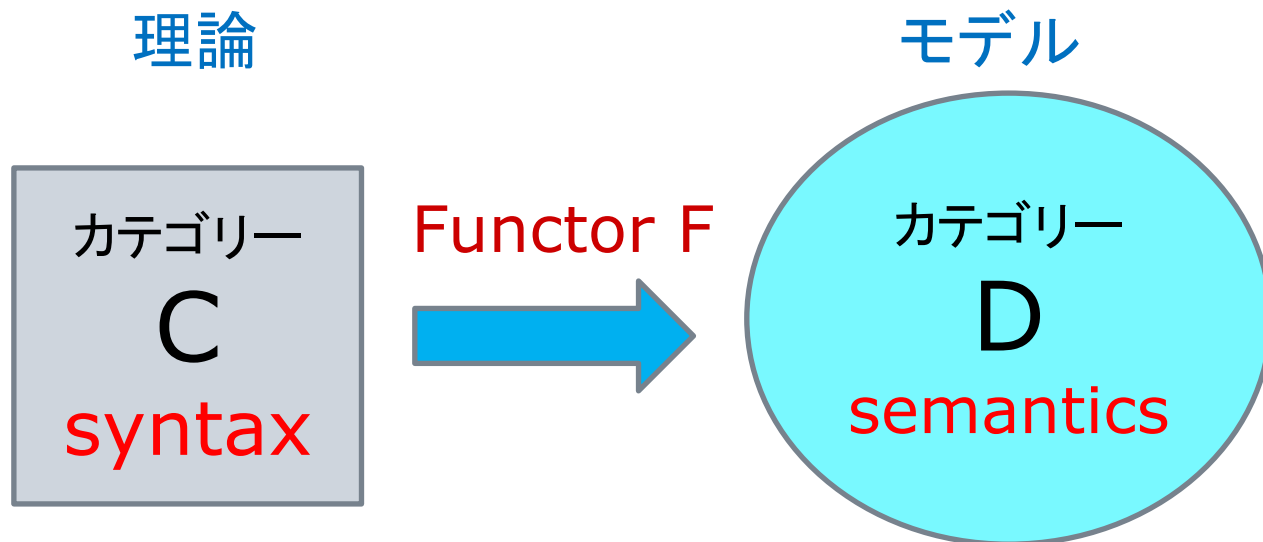
ローベールは、カテゴリーCからカテゴリーDへの、構造を保存するようなFunctor Fは、Dの中でCのある「解釈」を与えることに気づいた。それは、

Cが記号の結合方法を"**Syntax**"として記述し、
Dは、その"**Semantics**"を与えるということである。

$$F \\ C \rightarrow D$$

LawvereのFunctorial Semantics

Functor F は、**Syntax** \rightarrow **Semantics** として働き、
Syntaxにそれが成り立つ「意味」を持つ世界の「モデル」を与える。



サンプル – 群の表現

Dihedral group (二面体群)

次のように定義される数学的な構造 D_3 を考えよう。

$$D_3 = \langle r, s \mid r^3 = s^2 = rsrs = 1 \rangle$$

これだけだと、これがどんな構造かはよくわからない。

D_3 の r, s に次のような R と S を対応させる。(Functorだ！)

$$r \mapsto R = \begin{bmatrix} \cos(2\pi/3) & -\sin(2\pi/3) \\ \sin(2\pi/3) & \cos(2\pi/3) \end{bmatrix} \quad s \mapsto S = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

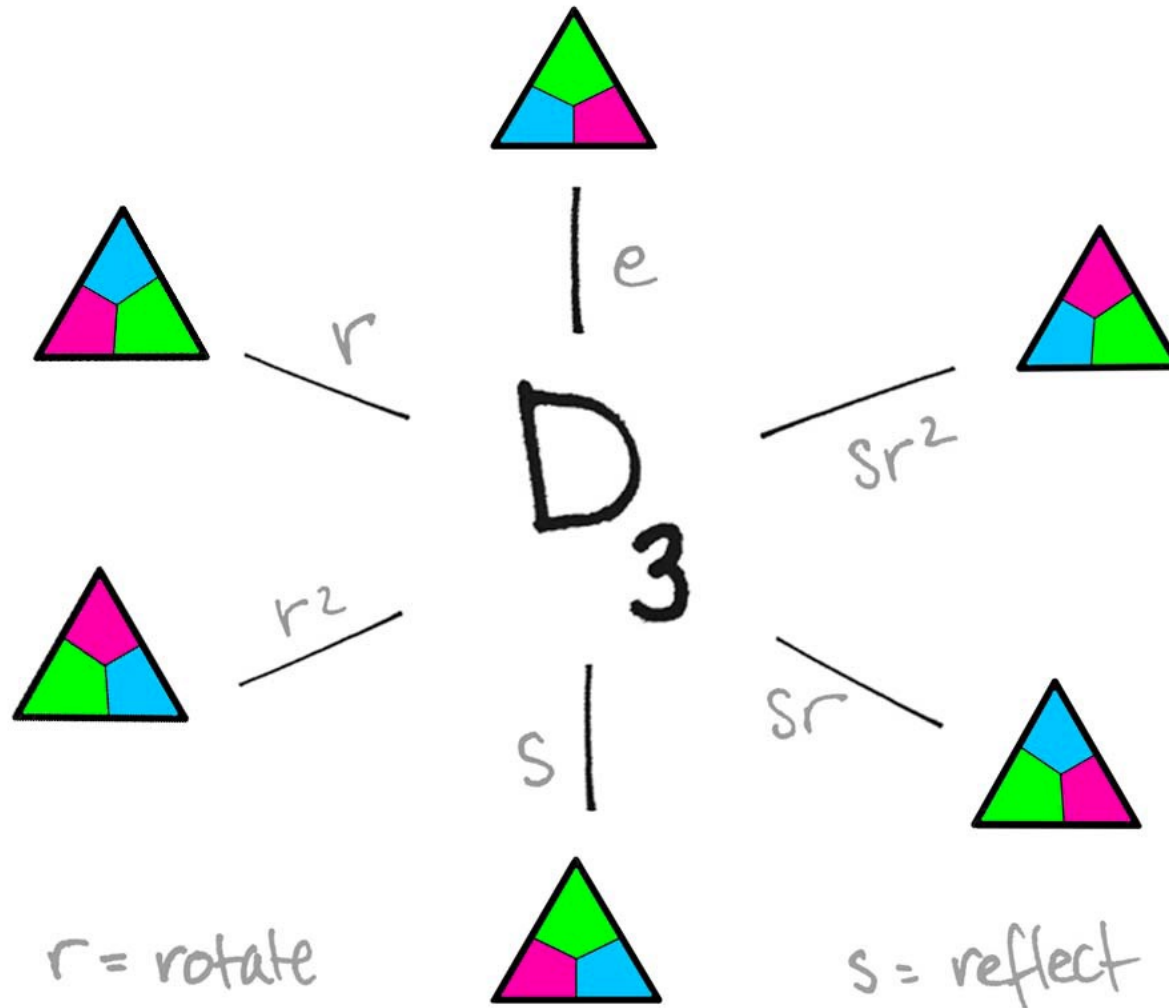
R は、左回りに60度の回転、

S は、垂直軸に沿って180度の回転。裏返しにする。

これでも、まだ分かりづらいかもしれない。

次の図を見れば、対応ははっきりする。

$$D_3 = \langle r, s \mid r^3 = s^2 = rsrs = 1 \rangle$$



“What is applied category theory?”
<https://arxiv.org/pdf/1809.05923.pdf>

“Quantum Computation and Lattice Problems” by Regev

We present the first explicit connection between quantum computation and lattice problems. Namely, we show a solution to the Unique Shortest Vector Problem (SVP) under the assumption that there exists an algorithm that solves the hidden subgroup problem on the dihedral group by coset sampling. Moreover, we solve the hidden subgroup problem on the dihedral group by using an average case subset sum routine.

<https://arxiv.org/abs/cs/0304005>

FUNCTORIAL SEMANTICS OF ALGEBRAIC THEORIES

AND SOME ALGEBRAIC PROBLEMS IN THE
CONTEXT OF FUNCTORIAL SEMANTICS OF
ALGEBRAIC THEORIES

<http://www.tac.mta.ca/tac/reprints/articles/5/tr5.pdf>

Functorial Semanticsについての Lawvereの“Author's comments”

数学は、多くの場面で、様々な抽象的な理論で計算を行うことで成り立っている。例えば、ある抽象的な理論をさらに別の理論に解釈するとか、あるいは、ある抽象的な理論を背景とした解釈で、より具体的な構造のカテゴリーを得るとか、また、これらのカテゴリーを、カテゴリー内やカテゴリー間でさらに構造を変形させるとか、等々。

ところで、1950年代には、ある種の構造に対して「カテゴリー」という言葉を使うことは、多くの数学者には、既に明らかであった。さらに、理論そのものや、その相互解釈がカテゴリーを形成する構造として考えることも、誰かがやろうと思えば明らかに可能であった。実際にHall, Halmos, Henkin, Tarskiなどは、既にその方向に大きく動いていたと思われる。

Functorial Semanticsについての Lawvereの“Author's comments”

しかし、「抽象的な理論」とその「具体的な背景」との関係そのものについてはどうであろうか。

二つのものを概念的に関連づけるためには、それらが共に共通のカテゴリーに属している必要がある。すなわち、数学的に言えば、両者が当初は異なるカテゴリーに属すると考えられていたのであれば、まず第一に、両者をFunctor的に、共通の第三のカテゴリーに移動させる必要がある。

しかし、もし両者を関連づけるこの試みが成功したならば、その時には、こうした関連付けすべての最も明快な説明には、両者自身がもともと、この第三のカテゴリーの「市民」であったことを示すことを含まれているだろう。

Functorial Semanticsについての Lawvereの“Author's comments”

集合全体や空間全体のような「背景」の場合、それらがカテゴリーを形成していることは既に明らかである。ここでは基本的な構造は「構成」である。

論理的な理論の場合、それが支える最も基本的な構造は「置換」の操作である。それも、「構成」の一つの形態として見るのが、一番効果的である。

したがって、理論をカテゴリーと見なすならば、モデルはファンクターなのだ!

Doctrine

John Baez <https://goo.gl/u63cvK>

Universal algebra began as an attempt to deal with lots of familiar algebraic gadgets simultaneously: groups, rings, lattices, vector spaces, Lie algebras, and so on. A bunch of the same theorems hold for *all* of these - so why not prove them all at once and be done with it? It turns out to be possible!

In his famous 1963 thesis, **Bill Lawvere** showed how universal algebra can be formulated using categories:

F. William Lawvere, ***Functorial Semantics of Algebraic Theories***.

He saw that universal algebra was all about algebraic gadgets that make sense in any category with finite products. Shockingly, every such category C can be seen as the “theory” of some such gadget. If D is some other category with finite products, a product-preserving functor $F:C\rightarrow D$: then gives such a gadget in D . We call this a **model** of C in D .

However, besides categories with finite products, there are also other kinds of categories, which support other kinds of algebraic gadgets. So, around 1969, Lawvere massively generalized universal algebra - so much that he required a precise definition of “a kind of category”! He called such a thing a “doctrine”:







第二部

ことばの構成性 — 文法 —

カテゴリー論と意味の形式的理論

ことばと意味の「構成性」について

第二部 ことばの構成性

-- 文法 --

- Minimalist Program
- ChomskyとLambek
- Combinatory Categorical Grammar
- Pregroup Grammar

Minimalist Program



カテゴリー論と意味の形式的理論

20TH ANNIVERSARY EDITION

The
Minimalist
Program の言語像



Noam
Chomsky

<https://goo.gl/ny5rSM>

Chomsky "Syntactic Structures"

言語に対する現代の計算主義的なアプローチは、人工知能研究と同じくらい古い。Church-Turingによる「計算可能性」の定式化やTuringによる「機械は考えることができるか？」という問題提起が行われた時代の1950年代に、Chomskyは登場する。Chomsky Hierarchyの論文が出たのは1956年、“Syntactic Structures”が出るのは1957年だ。

Chomsky “Minimalist Program”

1995年、Chomskyの“Minimalist Program”が出版される。それは、それまでの彼自身の言語理論をも一新したものだ。それは、それ以降の文法理論に深い影響を与えることになる。

“Minimalist”の名前は、必要最小限の構成要素から文法理論を再構成しようというアプローチを示すものだが、その直接のソースは、“Principles of minimal computation”（「最少計算の原理」）から来ている。彼は、最も簡単な計算操作は、二つのものを一つに合わせる“Merge”だとする。

言語能力を計算能力として捉えようとするときに、その中心的な課題は、文法的に正しい文を生成する計算規則を見つけることだ。それは、文法の計算ルールを見つけることだと言って良い。MinimalistのChomskyは、それをMergeだとする。

Mergeとは何か？

“The Science of Language: Interviews with James McGilvray” から

<https://goo.gl/lbAO7n>

Mergeとは何か？ 人間の進化史の中で

ある人の上に、その子孫にずっと伝えられる何か起きた。
あきらかに非常に短い間に、その遺伝子の変化はそのグループ内で優勢になった。それは、淘汰上の何らかの有利さがあったに違いない。ただ、それは非常に短い時間に、血縁上の小さなグループで起きた。さて、それは、なんだったのだろうか？

もっとも単純な想定は、それを疑う理由は何もないのだが、我々に起きたこととは、我々は Mergeを獲得したということだ。人は、すでに構成された心の中のオブジェクトをとって、それから、さらに大きな心の中のオブジェクトを構成することを可能にする操作を獲得したのだ。それがMergeだ。それを獲得するやいなや、人は、利用可能な表現や思考の階層的構造の無限の多様性をもつことになる。

Mergeとは何か？

感覚・運動システムと思考システムの先在

Mergeが導入された時、我々は、すでに感覚・運動システムを持っていた。おそらくそれは、周辺的なものとして利用された。事実、それを外部化しようという考えは、もっとあとになってうまれたに違いない。

さらに、我々は、ある種の思考のシステムを持っていた。しかし、それは原始的なもので、物事をなんらかのしかたで心に思い描くといったようなものだった。たとえそれがどんなものであれ、それらは動物のシステムと似たものには見えない。それが、17,18世紀によく議論された理由だ。ただ、あきらかに、それは存在していた。

Mergeとは何か？

人間の思考と言語の誕生

ひとたび、人がこの構成の技術と、これら(Chomskyが、「概念・思考システム」と呼ぶ思考システム)を使った、階層的に構造化された表現の無限の多様性を獲得すると、人は、突然、それまで誰もできなかったやり方で、考え、計画し、解釈することができるようになった。...

そして、この多様な思考を、なんとかして外部化しようという考えが生まれる。... そうしたものとして、言語の進化を捉えることができる。

Mergeとは何か？

“Minimalist program” wiki Page

ここでは、Mergeとはどういうものであるのか、簡単な例に基づいて述べる。

<https://goo.gl/x7Cyx6>

Merge

Merge は、二つのオブジェクト (α と β としよう) をとって、ラベルを持った (α か β である。この例では、 α がラベルになっている) 一つの順序を持たない集合に併合する関数である。ラベルは、この句 (phrase) の性質を表す。

$$\text{Merge} (\alpha, \beta) \rightarrow \{\alpha, \{\alpha, \beta\}\}$$

例えば、Merge は、Lexical Item “drink” と “water” に作用して、“drink water” を与える。

句 “drink water” の性質

この句 “drink water” は、名詞 *water* のようにではなく、動詞 *drink* のように振る舞うことに注意すること。

次の例のように、*drink* が置ける位置に、*drink water* を置くことができる。

- I like to _____ (*drink*)/ (*drink water*).
- (*Drinking/Drinking water*) _____ is fun.

さらに、句 *drink water* は、名詞 *water* と同じ場所には置くことができない

- There's some *water* on the table
- *There's some *drink water* on the table

ラベルの選択

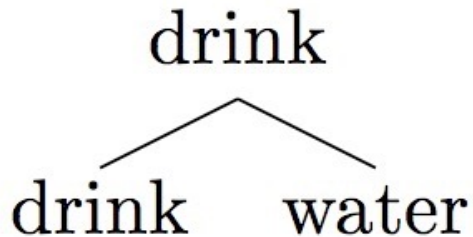
Merge (drink, water) の場合

句は、ラベルと同じものとみなされる。“drink water”の例では、この句は動詞として振舞うので、ラベルは、drinkになる。

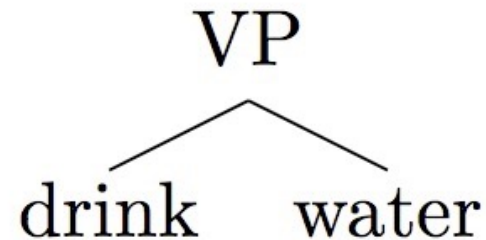
Merge (drink, water)

→ **{drink, {drink, water}}**

この句は、動詞句 (VP: Verb Phrase) とよばれる。構文木で表すと、次のように表現できる。



あるいは



ラベルの選択 Merge (cool, water) の場合

今度は、“cool”と“water”をMergeすることを考えよう。この時、この句 “cool water”は、動詞drink のようにではなく、名詞waterのように振る舞うことがわかる。

次の例のように、waterが置ける位置に、cool waterを置くことができる。

- I like to drink _____ (water)/(cool water).
- Drinking _____ (water)/(cool water) is fun.

“cool water”のラベルはwaterで、名詞句(NP)である。

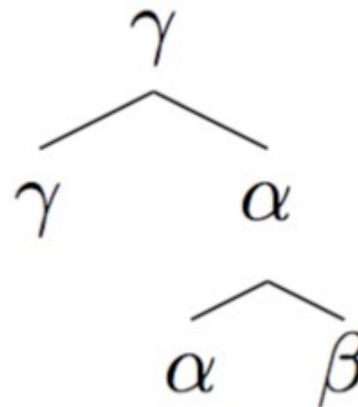
Merge (cool, water)
→ {water, {cool, water}}

既に構成された句とのMerge

Mergeは、既に構成されたものにも適用できる。それができなかつたら、こうしたシステムは、文法的には、二語文しか生成できないだろう。もし新しいhead γ が、既に構成されているオブジェクト(句だ)とMergeされれば、次のようになる。

Merge ($\gamma, \{\alpha, \{\alpha, \beta\}\}$)
→ **$\{\gamma, \{\gamma, \{\alpha, \{\alpha, \beta\}\}\}$**

次のような構文木になる。
ここで、 γ はラベルである。



Language = Interfaces + Recursion

計算システムとしての言語能力

Interfaces + Recursion = Language?

Chomsky's Minimalism and the View from Syntax-Semantics

Ull Sauerland
Hans-Martin Gärtner
(Editors)

SGG 89

<https://goo.gl/i0SVEP>

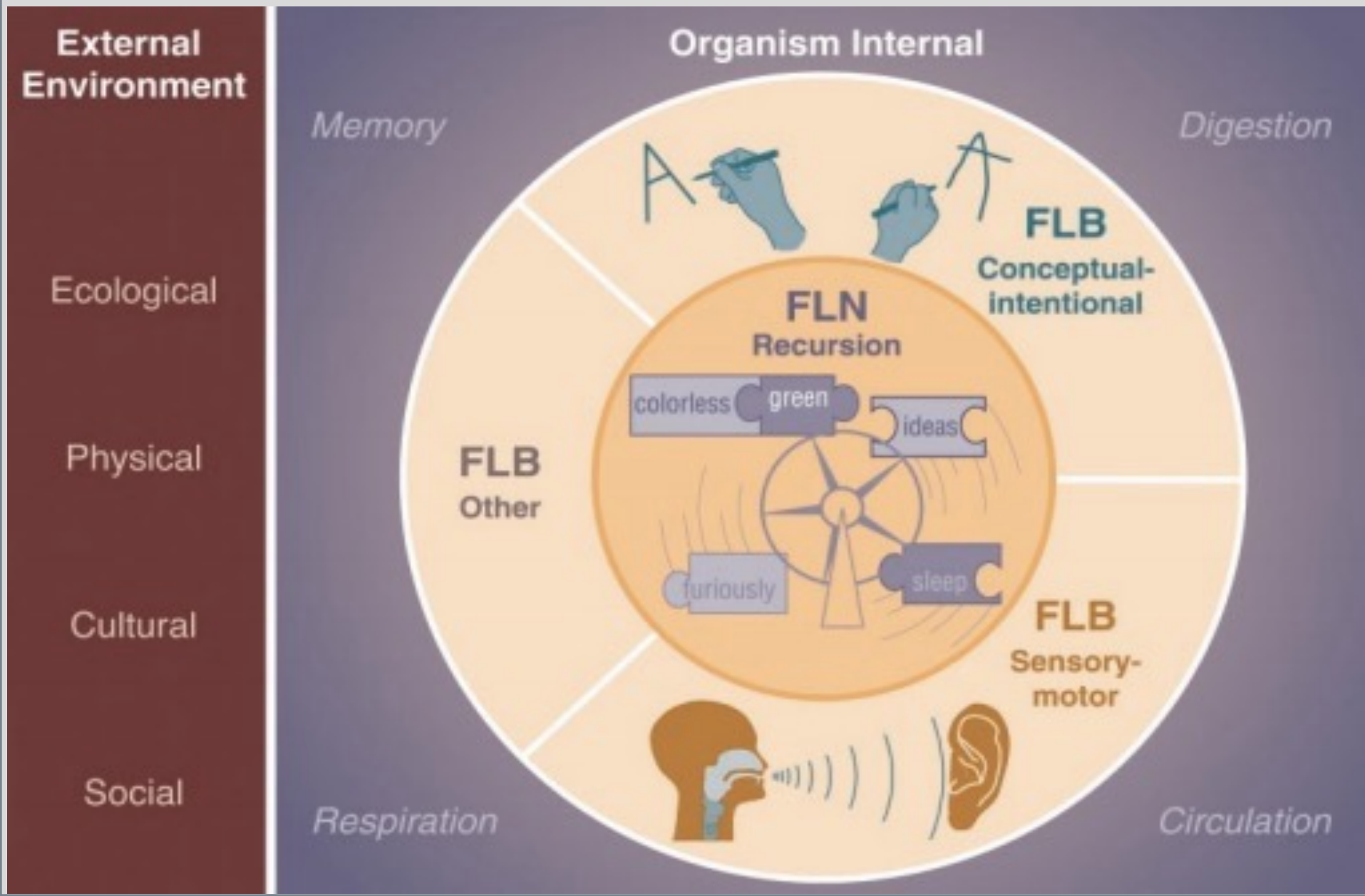
「計算システム」としての言語能力

Minimalist Programでは、言語能力を、二つのインターフェースをもつ計算システムとして捉える。

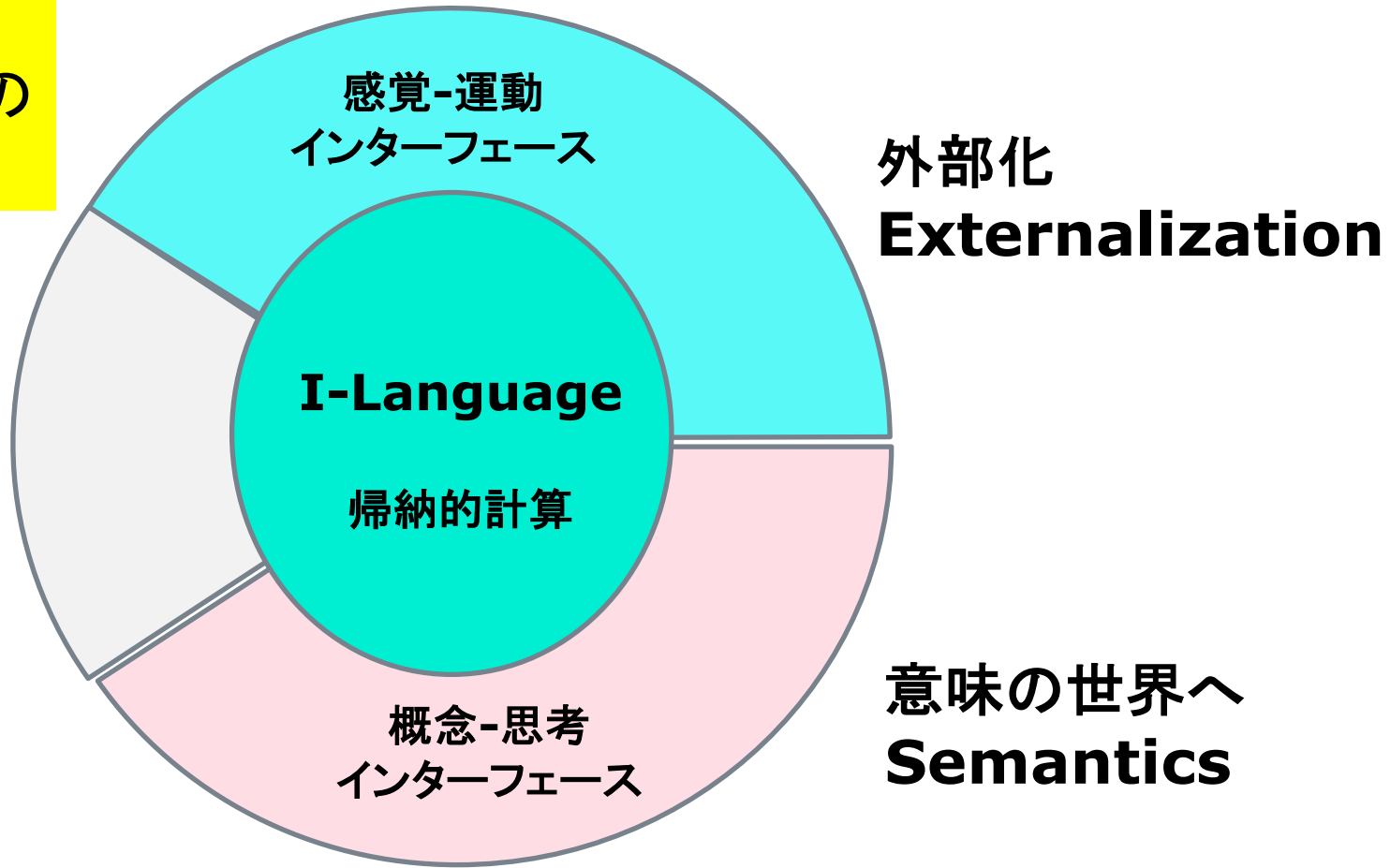
こうしたアプローチを、ボトムアップから積み上げてゆくということになれば、それは、機械による言語能力の実装に、近ずいてゆくことになるだろう。

ただ、そうした実装を、僕は知らない。

"The Faculty of Language: What Is It, Who Has It, and How Did It Evolve?" <http://goo.gl/pgSppH>

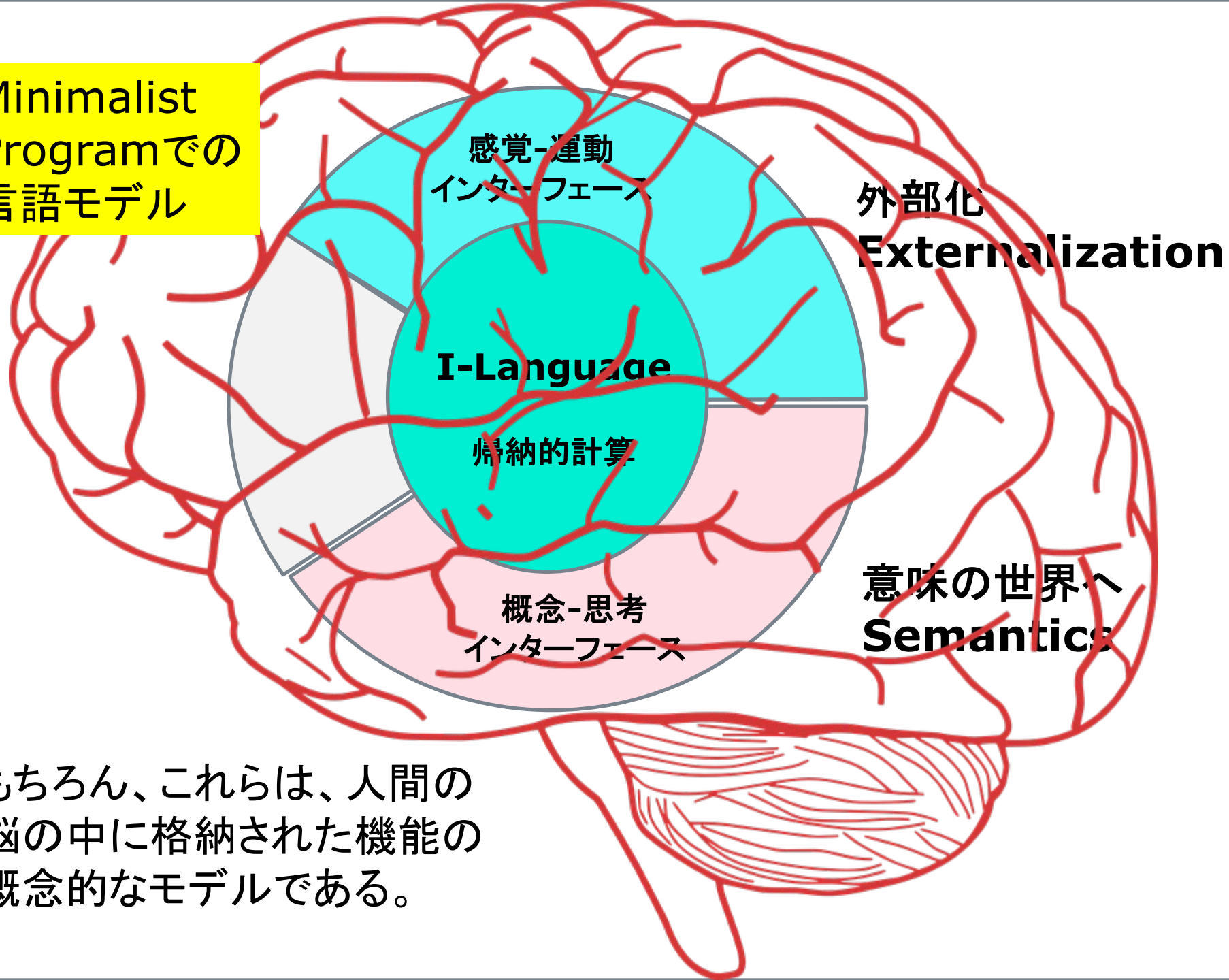


Minimalist
Programでの
言語モデル



Language = Interfaces + Recursion

Minimalist
Programでの
言語モデル



もちろん、これらは、人間の
脳の中に格納された機能の
概念的なモデルである。

Minimalist
Programでの
言語モデル

重要なことは、特に
概念-思考インター
フェースの先に、
言語能力を支える、
「概念のリソース」
とでもいうべきもの
が存在していること
である。



外部化
Externalization

感覚-運動
インターフェース

Phonetic form

I-Language

帰納的計算

Logical Form

概念-思考
インターフェース

概念
Semantics



概念のリソースの存在

その理由は、いろんなところで議論してきたし、ここでは、その問題は脇に置こう。ただ、もしそうなら、言語能力の遺伝的に決定された、さらなるコンポーネントが存在する(すなわち、その上で、言語能力が描く概念のリソースが存在する)。

そして、言語の進化の研究が取り組むべき問題が存在する。(すなわち、言語以前の、人間にも利用可能な概念のリソースの進化についてだ)

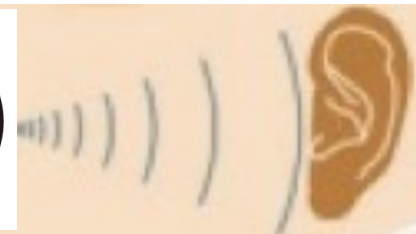
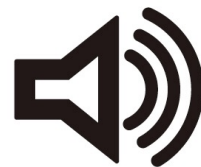
CIインターフェースの先にあるもの

Bot

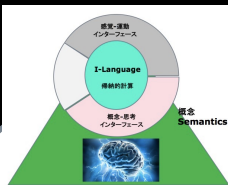
Botが行っていることは、言語モデル的には、こんなこと。

ただし、Bot内の小人の言語能力はフルに活用される。

もちろん、それはBotではなく、人間の能力である。



Text2Speech
Speech2Text



「中国人の部屋」と
同じ構造をしている



ChomskyとLambek



カテゴリー論と意味の形式的理論



Noam Chomsky

1928 –



Joachim Lambek

1922 – 2014

Chomsky 1957年 ”Syntactic Structures”

Noam Chomsky
Syntactic
Structures

mouton

de gruyter

言語に対する計算主義的なアプローチは、人工知能研究と同じくらい古い。Church-Turingによる「計算可能性」の定式化やTuringによる「機械は考えることができるか？」という問題提起が行われた時代の1950年代に、Chomskyは登場する。

Chomsky Hierarchyの論文が出たのは1956年、言語学に革命をもたらした”Syntactic Structures”が出るのは1957年だ。

Lambek 1958年

”The mathematics of sentence structure”

The mathematics of sentence structure*

JOACHIM LAMBEK

The definitions [of the parts of speech] are very far from having attained the degree Euclidean geometry.

Otto Jespersen ([1924](#)).

Contents

1. [Introduction](#)
2. [Syntactic types](#)
3. [Type list for a fragment of English](#)
4. [Formal systems](#)
5. [Type computations in English](#)
6. [Pronouns](#)
7. [Syntactic calculus](#)
8. [Decision procedure](#)
9. [Proof of Gentzen's theorem](#)
10. [Algebraic remarks](#)
11. [NOTES](#)
12. [REFERENCES](#)

Chomskyの僚友だったLambekは、驚くべき発見をする。

文法の計算ルールは、次のたった二つの式で表されるというのだ。

$$(x/y)y \rightarrow x$$

$$y(y \setminus x) \rightarrow x$$

Lambekは、名詞を表す型 n と、文を表す型 s というたった二つの型を用いて、語の並びから、先の二つの計算ルールで文を導く計算を試みせる。

Categorial Grammarの誕生である。

Lambekの回想から

「その数年後、私はアメリカ数学会のシンポジウムに招かれ、言語学への数学の応用について話をした [1961]。新しいことを言うのは難しいので、私は $(xy)z = x(yz)$ という結合律が成り立たないような、構文計算について話した。

しかし、こうした考えは言語学のコミュニティからは受け入れられなかった。それどころか、ある大学院生は、敵意剥き出しの反応をしたほどだった。ただ、ノーム・チョムスキーとロバート・リースが、私に研究を続けるようにと励ましてくれた。それでも私は、チョムスキーの生成-変形文法という新しい波が他のすべてを一掃するのを目の当たりにし([Chomsky's generative-transformational grammar sweeping everything else aside](#))、他のことに目を向けるようになった。」

“Remarks on the history of categorial grammar”から

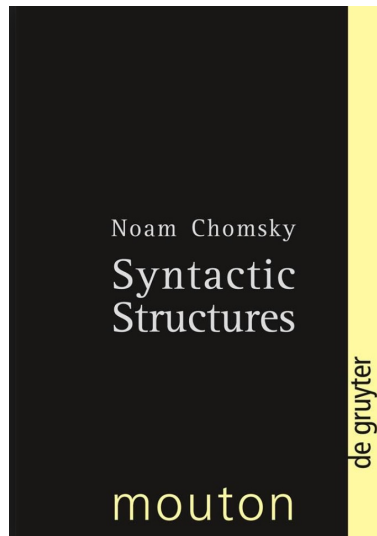
Chomsky, Lambek 再び出会う

1997年、Chomskyは“Minimalist Program”を発表して、それまでの理論を一新する。

1998年、LambekはCategorial Grammarの新しい定式化 Pregroup 文法を発表する。2008年には、Lambekは、それらをまとめた“From Word to Sentence”を発行する。

40年以上の時を隔てて、両者は再会する。

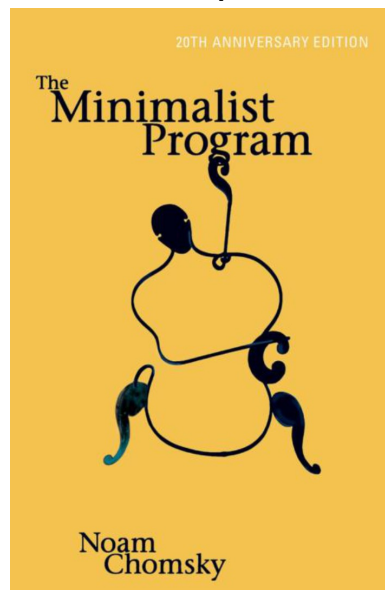
1957年



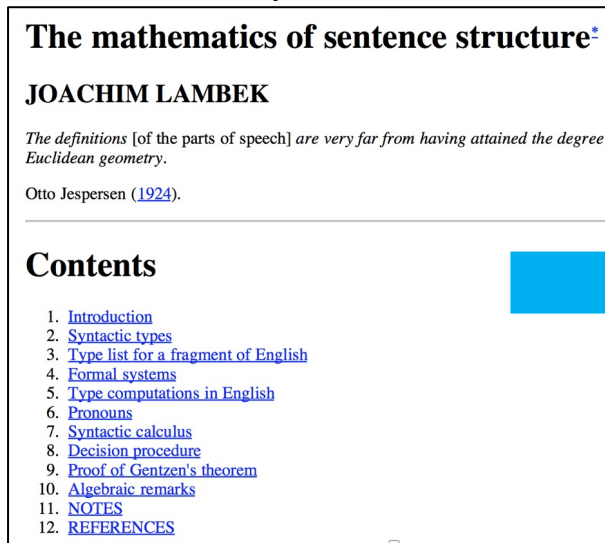
Chomsky



1995年



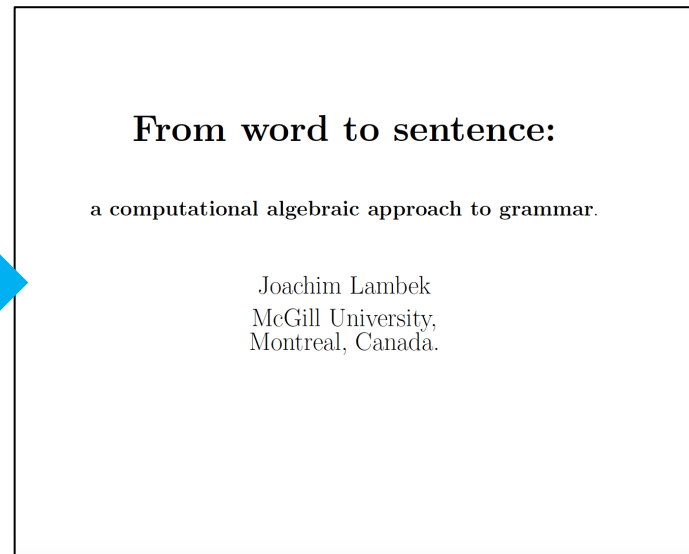
1958年



Lambek



2008年



Minimalist Programと Pregroup Grammar

「現在の言語理論に詳しい読者であれば、このモノグラフの根底にある多くの考え方が、過去半世紀にチョムスキーとその学派によって得られた深い洞察に影響を受けていることにお気づきだろう。」

「チョムスキーの思想は長年にわたって大きな変化を遂げ、その結果、多くの弟子たちも同様に方法論を見直すことになるのが常である。... 私はこれらの理論をすべてマスターしているわけではないが、そのうちのいくつかはpregroupの枠組みに取り入れることができるのではないかと思っている。」

「私はミニマリストのプログラムを完全に理解したとは言えないが、技術的な詳細はともかく、精神的にはpregroup文法はそのようなプログラムを実行する試みと見なすことができると考えている。」

Minimalist文法とカテゴリー文法の収斂について

1 AT THE CARTESIAN WELL: THE MINIMALIST PROGRAM & CATEGORIAL GRAMMAR

Imagine the following scene. You are at your favorite beer hall somewhere in Amsterdam—let's call it the Cartesian Well. Well known meeting place of intelligentsia, you are not surprised when a thin person dressed all in black sidles up to you and whispers in your ear, "Have I got a linguistic theory for you!" You of course yawn, have heard many such fables in your time; besides you have drunk too much. "No, wait," the figure grabs your shoulder, "I've discovered that Chomsky's latest approach to syntax and categorial grammar are *converging*."

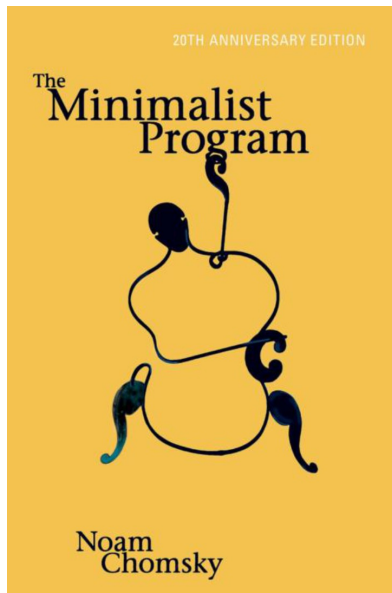
Another flat-earth cultist? you think. "Well, hear me out—at least let me buy you another beer." So you to listen to the tale:

4 CONCLUSIONS FROM THE BEER HALL

To summarize, given the current push towards "minimalism" in the so-called government-and-binding approach seems to have eliminated both government (and binding, not discussed here) in favor of a single hierarchical concatenation operator that meshes perfectly with the classical theory of categorial grammar, as well as providing a natural *explanation* for the observed grammatical relations and a transparent framework on which to build a model of sentence processing. While this trend surely does not solve all our "religious" problems, it certainly goes a long way towards taking down the "barriers" to a mature, ecumenical framework within which to reach common ground among what has long appeared to be quite disparate accounts of natural language. Perhaps we can all now drink beer together.

1995年 Berwick & Epstein

"On the Convergence of 'Minimalist' syntax and Categorical Grammar"



1995年

On the Convergence Of 'Minimalist' Syntax and Categorial Grammar

Professor Robert C. Berwick, MIT Department of Computer Science

Professor Samuel David Epstein, Harvard Department of Linguistics
Cambridge, MA 02139

ABSTRACT

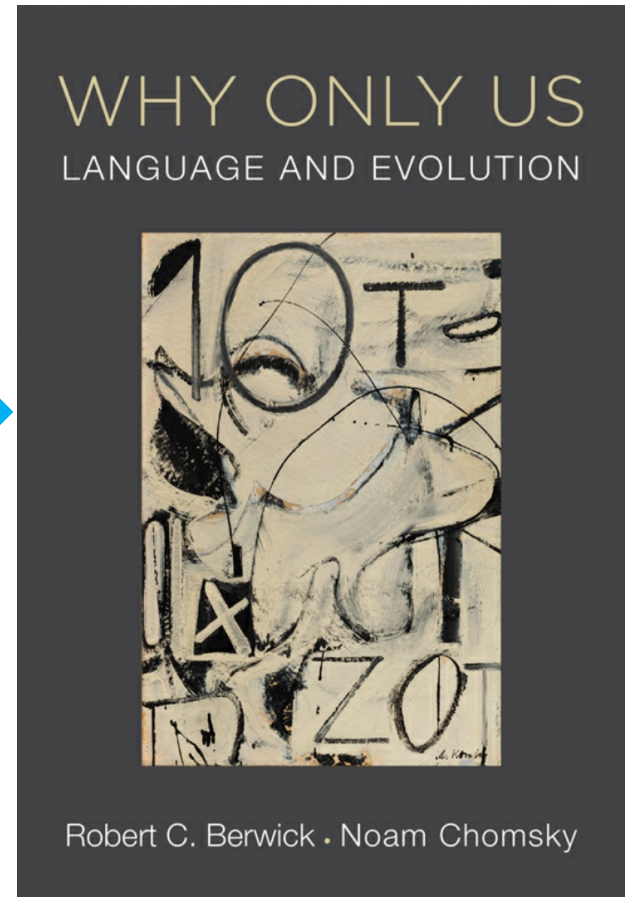
This paper shows that the so-called "Minimalist Program" of Chomsky (1993, 1995) can be given a natural interpretation as a categorial system in which there is exactly one syntactic (algebraic) operation: namely, "Hierarchically Concatenate" (HC) (what Chomsky calls "Merge"), and also replacing the representations of "D-structure," "S-structure" and transformations with the derivation lines typical of categorial systems—thus unifying two previously disparate approaches to the analysis of natural language. For example, the general "movement" rule of transformational grammar is easily seen to be a subcase of Hierarchical Concatenation of (α , β), where α is a subtree of β ; this automatically derives the usual c-command condition on so-called "empty categories." The usual semantic interpretation benefits of the categorial

interpret" sentences.

1 AT THE CARTESIAN WELL: THE MINIMALIST PROGRAM & CATEGORIAL GRAMMAR

Imagine the following scene. You are at your favorite beer hall somewhere in Amsterdam—let's call it the Cartesian Well. Well known meeting place of intelligentsia, you are not surprised when a thin person dressed all in black sidles up to you and whispers in your ear, "Have I got a linguistic theory for you!" You of course yawn, have heard many such fables in your time; besides you have drunk too much. "No, wait," the figure grabs your shoulder, "I've discovered that Chomsky's latest approach to syntax and categorial grammar are *converging*."

2016年



Berwick & Chomsky

Combinatory Categorical Grammar



カテゴリー論と意味の形式的理論

カテゴリー文法の「カテゴリー」とは

カテゴリー文法の「カテゴリー」は、数学のカテゴリー論の「カテゴリー」とは、直接の関係はない。

それは、従来の文法の「品詞(Part of Speech)」概念の拡大である。その拡大された「品詞」をカテゴリーと呼んでいる。

このカテゴリーは、語の接続によって引き起こされる計算上の「型」を持つ。

The mathematics of sentence structure

JOACHIM LAMBEK

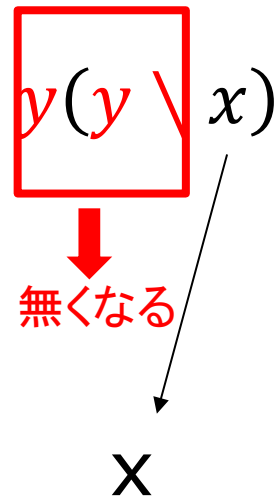
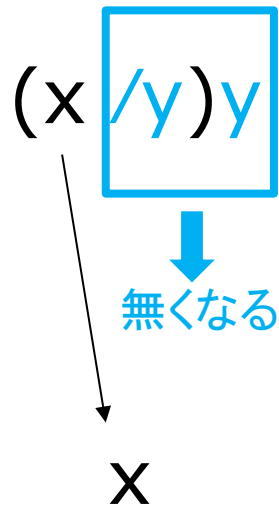
http://lecomte.al.free.fr/ressources/PARIS8_LSL/Lambek.pdf

Syntactic types

- 二つのPrimitive Type
 - 文を表す型 s
 - 名詞を表す型 n
- 複合型 Compound Type はリカーシブに定義される
 - X と y が型なら、 x / y も $x \setminus y$ も型である。
 - それぞれ、 x over y , y under x と読む。

Syntactic types

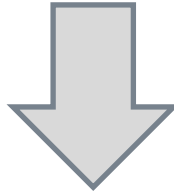
$$(I) \quad (x/y)y \rightarrow x, \quad y(y \setminus x) \rightarrow x.$$



Syntactic types

$$(I) \quad (x/y)y \rightarrow x, \quad y(y \setminus x) \rightarrow x.$$

1958年の定式化



2008年の定式化

$$xx^r \rightarrow 1, \quad x^l x \rightarrow 1,$$

Type list for a fragment of English

いくつかの英文の例で、型を調べてみよう

Sample 1
John works

John Works

$n \quad n \setminus s$



$n(n \setminus s)$

$n(n \setminus s)$



s

Sample 2
Poor John works

(Poor John) works

$n/n \quad n \quad n \setminus s$



$(n/n)n(n \setminus s)$

$(n/n)n(n \setminus s)$

$n(n \setminus s)$



s

Sample 3
John works here

(John works) here

$n \quad n \setminus s \quad s \setminus s$



$n(n \setminus s) (s \setminus s)$

$n(n \setminus s) (s \setminus s)$

$s(s \setminus s)$



s

Sample 4
John never works

John (never works)

$n \quad (n \setminus s) / (n \setminus s) \quad n \setminus s$



$n((n \setminus s) / (n \setminus s)(n \setminus s))$

$n((n \setminus s) / (n \setminus s)(n \setminus s))$

$n(n \setminus s)$



s

Sample 5
John works for Jane

(John works) (for Jane)
 $n \quad (n \setminus s) \quad (s \setminus s) / n \quad n$



$(n(n \setminus s)) \quad ((s \setminus s) / n)n$
 $(n(n \setminus s)) \quad ((s \setminus s) / n)n$

$s(s \setminus s)$



s

Sample 6

John works and Jane rests

(John works) (and (Jane rests))
 $(n \ (n \setminus s)) \ ((s \setminus s) / s \ (n \ (n \setminus s)))$



$(n(n \setminus s))((s \setminus s) / s(n(n \setminus s)))$
 $(n(n \setminus s))((s \setminus s) / s(n(n \setminus s)))$

$s(s \setminus s) / s(s)$



s

Sample 7
John likes Jane

John (likes Jane)

$n \ ((n \setminus s) / n \ n)$



$n \ ((n \setminus s) / n \ n)$

$n \ ((n \setminus s) / n \ n)$

$n \ (n \setminus s)$



S

Sample 7'
John likes Jane

(John likes) Jane
(n n \ (s / n)) n



(n n \ (s / n)) n
(n n \ (s / n)) n
s / n n



s

sample7とsample7'

Sample7

John (likes Jane)
 $n \quad ((n \setminus s) / n \quad n)$

Sample7'

(John likes) Jane
 $(n \quad n \setminus (s / n)) \quad n$

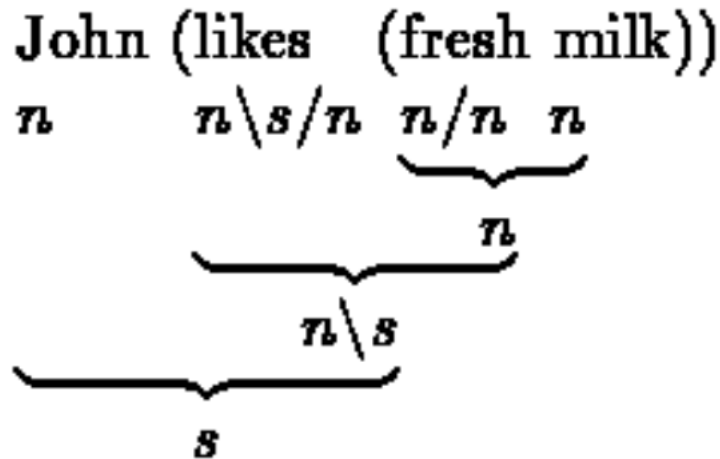
$$(n \setminus s) / n \iff n \setminus (s / n)$$

$$(II) \quad (x \setminus y) / z \iff x \setminus (y / z).$$

語の型と品詞

	Word	Type	Part of Speech	:
(1)	<i>works</i>	$\tau \backslash s$	自動詞	
(2)	<i>poor</i>	τ / τ	形容詞	
(3)	<i>here</i>	$s \backslash s$	副詞	
(4)	<i>never</i>	$\tau \backslash s / (\tau \backslash s)$	副詞	
(5)	<i>for</i>	$s \backslash s / \tau$	前置詞	
(6)	<i>and</i>	$s \backslash s / s$	接続詞	
(7)	<i>likes</i>	$\tau \backslash s / \tau$	他動詞	

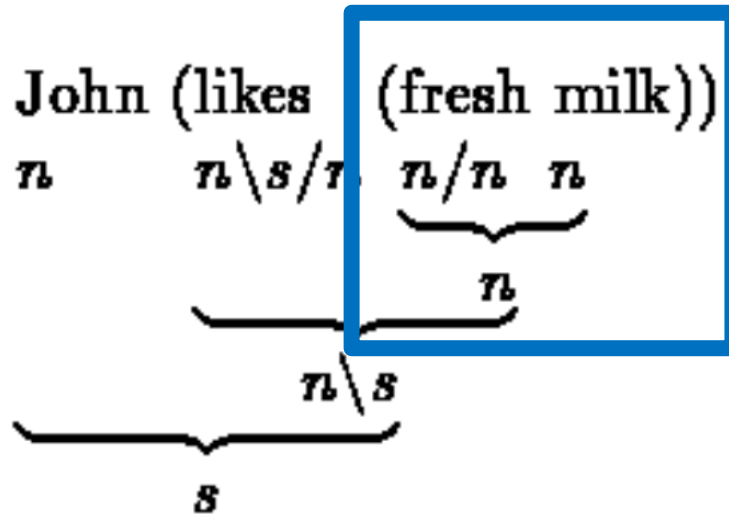
Formal systems



文の文法性の計算は、隣り合う語のカテゴリーの型の計算から、文のカテゴリー s を導出する形式的体系で記述できる。

$$\tau_b((\tau_b \backslash S / \tau_b)((\tau_b / \tau_b)\tau_b)) \rightarrow \tau_b((\tau_b \backslash S / \tau_b)\tau_b) \rightarrow \tau_b(\tau_b \backslash S) \rightarrow S.$$

Formal systems



Merge (cool, water)
 $\rightarrow \{\text{water}, \{\text{cool}, \text{water}\}\}$
 Mergeの議論での上のような導出を思い出して欲しい。同じことだ。ただし、ここでは“water”というラベルは不要である。

$$T_b((T_b \setminus S / T_b)((T_b / T_b) T_b)) \rightarrow T_b((T_b \setminus S / T_b) T_b) \rightarrow T_b(T_b \setminus S) \rightarrow S.$$

Type computations in English

- i. 許されるすべてのやり方で括弧を挿入する。
- ii. 各単語に、与えられた有限の型リストで許されるすべての型を割り当てる。(「and」や「only」のように無限の型を持つ可能性のある単語から生じる困難はひとまず無視する)。
- iii. それぞれのグルーピングと型の割り当てに対して、式全体の型を計算する。
- iv. グルーピングと型の割り当てのうち、望ましい型 s をもたらしものを選択する。

Pronouns

(8) he works, he likes Jane
 $s/(\tau \backslash s)$ $\tau \backslash s$ $s/(\tau \backslash s)$ $\tau \backslash s/\tau$ τ

(9) that's him, Jane likes him,
 s/τ $(s/\tau) \backslash s$ τ $\tau \backslash s/\tau$ $(s/\tau) \backslash s$

Jane works for him
 τ $\tau \backslash s$ $s \backslash s/\tau$ $(s/\tau) \backslash s$

(10) he likes him ;
 $s/(t \setminus s) \quad t \setminus s/t \quad (s/t) \setminus s$

$(s/(t \setminus s))(t \setminus s/t)((s/t) \setminus s)$

(III) $(x/y)(y/z) \rightarrow x/z, \quad (x \setminus y)(y \setminus z) \rightarrow x \setminus z.$

$(s/(t \setminus s))(t \setminus s/t) \rightarrow s/t$

$(t \setminus s/t)((s/t) \setminus s) \rightarrow t \setminus s$

$\underbrace{(\text{he likes})}_{s/\tau_b}$ him, he $\underbrace{(\text{likes him})}_{\tau_b \setminus s}$
 $(s/\tau_b) \setminus s$ $s/(\tau_b \setminus s)$

(John works) for Jane
 τ_b $\tau_b \setminus s$ $s \setminus s/\tau_b$ τ_b

$(\tau_b \setminus s)((s \setminus s/\tau_b)\tau_b) \rightarrow (\tau_b \setminus s)(s \setminus s) \rightarrow \tau_b \setminus s.$

$\tau_b \rightarrow s/(\tau_b \setminus s), \quad \tau_b \rightarrow (s/\tau_b) \setminus s$

$$(IV) \quad \mathfrak{x} \rightarrow \mathfrak{y}/(\mathfrak{x}\backslash\mathfrak{y}), \quad \mathfrak{x} \rightarrow (\mathfrak{y}/\mathfrak{x})\backslash\mathfrak{y}.$$

$$\tau_b^* (\tau_b^* \backslash \tau_b^* / \tau_b') \tau_b \rightarrow (\tau_b^* / \tau_b') \tau_b \rightarrow (\tau_b^* / \tau_b') \tau_b' \rightarrow \tau_b^*$$

Pregroup Grammar



カテゴリー論と意味の形式的理論

From word to sentence: a computational algebraic approach to grammar

Joachim Lambek

<http://www.math.mcgill.ca/barr/lambek/pdffiles/2008lambek.pdf>

“She will see him” の解析

She will see him

 | | | |

π_3 $\pi^r s_1 j^l$ io^l o

She : π_3
will : $\pi^r s_1 j^l$
see : io^l
him : o

先の例に出ている「基本的な型」

She will see him
 $\pi_3(\pi^r s_1 j^l)(i o^l) o$

π = 主語

π_3 = 三人称単数の主語

s_1 = 現在形の宣言文

i = 自動詞の不定形

j = 完全動詞句の不定形

o = 直接目的語

半順序 →

→ を、次の条件を満たす半順序とする。

反射率: $x \rightarrow x$

推移率: $x \rightarrow y$ で $y \rightarrow z$ なら、 $x \rightarrow z$

反対称律: $x \rightarrow y$ で $y \rightarrow x$ なら $x = y$

次のルールを認める。

$\pi_3 \rightarrow \pi, i \rightarrow j$

$x \rightarrow y$ は、型 x のすべてが型 y でもあると読むことができる。

文 s_1 の導出

ルール $\pi_3 \rightarrow \pi, i \rightarrow j$ のもとで、

$$\begin{aligned} & \textit{She will see him} \\ & \pi_3(\pi^r s_1 j^l)(i o^l) o \\ & = [\pi_3 \pi^r] s_1 [j^l i] [o^l o] \\ & \rightarrow [\pi \pi^r] s_1 [j^l j] [o^l o] \\ & \rightarrow 1 s_1 1 1 \rightarrow s_1 \end{aligned}$$

ここで、 $(xy)z = x(yz)$ 、 $1x = x = x1$ 、
 $xx^r \rightarrow 1, x^l x \rightarrow 1$ を使った。

She will see him

$$\pi_3(\pi^r s_1 j^l)(i o^l) o = [\pi_3 \pi^r] s_1 [j^l i] [o^l o]$$

$$\rightarrow [\pi \pi^r] s_1 [j^l j] [o^l o]$$

$$\rightarrow 1 s_1 1 1 = s_1,$$

$$(xy)z = x(yz)$$

$$xx^r \rightarrow 1, \quad x^l x \rightarrow 1,$$

$$1x = x = x1,$$

"she will come"

She will come

$\pi_3 \pi^r s_1 j^l i$

will: $\pi^r s_1 j^l$

$\pi_3 (\pi^r s_1 j^l) i$

$\rightarrow \pi (\pi^r s_1 j^l) i$

$\rightarrow \pi (\pi^r s_1 j^l) j$

$\rightarrow \pi \pi^r s_1 j^l j$

$\rightarrow 1 s_1 1$

$\rightarrow s_1$

"She will see him"

She will see him

$\pi_3 \pi^r s_1 j^l i o^l o$

$\pi_3 (\pi^r s_1 j^l) (i o^l) o$

$\rightarrow \pi \pi^r s_1 j^l i o^l o$

$\rightarrow \pi \pi^r s_1 j^l j o^l o$

$\rightarrow (\pi \pi^r) s_1 (j^l j) (o^l o)$

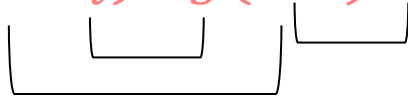
$\rightarrow 1 s_1 1 1$

$\rightarrow s_1$

“Will she see him?”

Will she see him ?

$(q_1 j^l \pi_l) \pi_3 (i o^l) o \rightarrow q_1$




will: $q_1 j^l \pi^l$

ここで、新しい型 q_1 を導入する
 q_1 は、yesかno で答える疑問文の現在形。
また、willに 型 $q_1 j^l \pi^l$ を割り当てる。

"Whom will she see?"

Whom will she see ?

$$(\bar{q}o^{ll}q^l)(q_1j^l\pi^l)\pi_3(io^l) \rightarrow \bar{q}$$


ここで、新しい型 q と \bar{q} を導入する

q は、yesかno で答える任意の時制の疑問文

\bar{q} は、wh-疑問文を含む任意の疑問文

$q_1 \rightarrow q \rightarrow \bar{q}$ である。

また、 $x^{ll}x^l \rightarrow 1$, $x^r x^{rr} \rightarrow 1$

(3.1)のような文が、
どのようにリアルタイムで解析されるのか？

$$(3.1) \quad \textit{whom will she see} - ?$$

$$(\bar{q}o^{ll}q^l)(q_1j^l\pi^l)\pi_3(io^l) \rightarrow \bar{q}$$

$$\textit{whom} : \bar{q}\hat{o}^{ll}q^l,$$

$$\textit{whom will} : \bar{q}\hat{o}^{ll}q^l q_1 j^l \pi^l \rightarrow \bar{q}\hat{o}^{ll}j^l \pi^l,$$

$$\textit{whom will she} : \bar{q}\hat{o}^{ll}j^l \pi^l \pi_3 \rightarrow \bar{q}\hat{o}^{ll}j^l,$$

$$\textit{whom will she see} : \bar{q}\hat{o}^{ll}j^l i o^l \rightarrow \bar{q}\hat{o}^{ll}o^l \rightarrow \bar{q}.$$







第三部

意味の構成性

-- カテゴリー論的分散意味論 --

カテゴリー論と意味の形式的理論

ことばと意味の「構成性」について

第三部 意味の構成性

-- カテゴリー論的分散意味論 --

- カテゴリー論的構成的分散意味論 Overview
- DisCoCat入門
- DisCoCat -- Coecke's original form
- Interlude -- Monoidal categoryとString Diagram
- DisCoCat -- Coecke's diagrammatic calculus

カテゴリー論的 構成的分散意味論 Overview



カテゴリー論と意味の形式的理論

カテゴリー論的 構成的分散意味論について

第三部では、カテゴリー論を用いた、構成的分散意味論を紹介する。

基本的な問題意識は、「文の意味は、個々の語の意味と、語から文を構成する文法規則によって構成的に決定される」という構成性の原理を文の意味論においても適用することである。

「カテゴリー論的構成的分散意味論」 **Distributional Compositional Categorical Semantics** を **DisCoCat** と呼ぶことがある。

DisCoCatを提唱した最初の論文は、Bob Coeckeらが2010年に発表した次の論文である。

Bob Coeckeの
Categorical QM については、彼の著書“Picturing
Quantum Processes”に、
彼のアプローチの特徴がよく
表われている。

彼が愛用するString
Diagramについては、丸山
のセミナー資料
「String Diagram を学ぶ --
カテゴリー論入門 (1)」を参
考にしてほしい。

<https://www.marulabo.net/docs/category01/>

PICTURING QUANTUM PROCESSES

A First Course in Quantum Theory and
Diagrammatic Reasoning

BOB COECKE AND ALEKS KISSINGER



カテゴリー論的構成的分散意味論の発展

Bob Coeckeらが始めた「カテゴリー論的構成的分散意味論」は、この10年の間に大きく発展した。

現在では、量子コンピュータを使った自然言語処理 **QNLP** (Quantum Natural Language Processing) という新しい分野で、活発に研究が続けられている。

「カテゴリー論」+「量子論」が、「構成的分散意味論」と結びつくのだ。今回のセミナーでも、こうした流れを紹介したいと思っている。

DisCoCat入門

DisCoCatの入門として、Tai-Danae Bradley の次の論文を紹介する。

What is applied category theory

3.2 Natural Language Processing

Tai-Danae Bradley

2018年

[https://arxiv.org/pdf/
1809.05923.pdf](https://arxiv.org/pdf/1809.05923.pdf)



Syntax = 文法
Semantics = 意味

「Syntax = 文法」と「Semantics = 意味」の対応づけに、**Functorial Semantics**のフレームを使うには、どのような準備が必要だろうか？ 次の三つが考えられる。

1. **文法のCategory**: 文法を数学的に扱うにはどうすればよいか？ 文法を数学的なCategoryとして捉える。
2. **意味のCategory**: 意味を数学的に扱うにはどうすればよいか？ 意味を数学的なCategoryとして捉える。
3. **Functor**: 文法 \rightarrow 意味 のFunctorをどのように構成するか？ このFunctorは、文全体の意味を捉えることができるか？

The Syntax Category:
Pregroup Grammar = PregX

文法のCategory PregX

文法のCategoryでは、先に紹介したLambekのPregroup Grammarを使う。(ただし、その簡略版である。)
このカテゴリーをPregXで表そう。

基本的な文法の型は、名詞の型 n と文の型 s である。これから、基本的な品詞の型が定義されていく。

$$\begin{aligned}n &= \text{名詞} \\s &= \text{文} \\nn^l &= \text{形容詞} \\n^r sn^l &= \text{他動詞}\end{aligned}$$

語が接続する時、対応する語の型で、次のような計算を行う。

$$\begin{aligned}ww^l &= \varepsilon^l \text{ (消える)} \\w^r w &= \varepsilon^r \text{ (消える)}\end{aligned}$$

"yellow banana" は名詞として扱われる

"yellow" is
an adjective



yellow

$n \ n^e$

"banana" is
a noun



banana

n

$1_n \cdot \varepsilon^e$
→

"yellow banana"
is a noun



yellow banana

$1 \cdot n = n$
 $n \cdot 1$

Success!

apply the
identity on n
here

apply the
counit ε^e here

“bananas are fruit” は文である

“bananas” is
a noun



bananas

“are” is
a verb

are

“fruit” is
a noun



fruit

“bananas are fruit”
is a sentence



bananas are fruit

n n^r s n^l n
 ───────── ───────── ─────────
 apply ϵ^r apply identity apply ϵ^l

$\epsilon^r \cdot 1_s \cdot \epsilon^l$
 ───────────────────────────────────→

S

$$nn^r sn^l n = nn^r \cdot s \cdot n^l n \xrightarrow{\epsilon^r \cdot 1_s \cdot 1_n} 1 \cdot s \cdot n^l n = s \cdot n^l n \xrightarrow{1_s \cdot \epsilon^l} s \cdot 1 = s$$

The Semantics Category:
Vector Spaces = FVec

意味の表現について

意味の表現については、二つの流儀がある。

- 論理的な形式を用いるもの
CCGでのCurry-Howard-Lambek対応を用いる
- ベクトル空間で表現するもの
今回のセミナーのDisCoCatがそうである

語の意味の分散モデル

puppy という語が、cute, furry, bark, pet ... というような後の近くに、よく現れるとしよう。別の例だと、yegg という語の意味をもしも知らなかったとしても、前後の文からその意味を推察できるかもしれない。

だから、ある語の意味をベクトルで表現できる。こうしたアプローチは、しばしば、**意味の分散モデル**といわれることがある。

しかし、正確には、**word** \mapsto **vect** という割り当ては、何を意味するのだろうか？

語の意味の分散モデルを作る

例えば好きな本でもいい、ある固定したコーパスがあったとしよう。そこから、context words $\{w_1, \dots, w_n\}$ と呼ばれる語の集まりを選ぶ。それは、コーパス内のすべての言葉でもいいし、その一部でもいい。この $\{w_1, \dots, w_n\}$ の w_i をベクトル空間 V の i 番目の標準的な基底と考えるのだ。

そうすれば、コーパス内のすべての語は、次のような context word の線形結合で表されるベクトル表現を持つことになる。

$$w = \sum_{i=1}^n c_i w_i$$

この係数 c_i は、コーパス内で、語 w が語 w_i の近くに現れた回数を示す実数である。

語の意味の分散モデルの例

{sweet, green, furry} という語を含むある本があったとしよう。この三つの語をこのコーパスのcontext word に選んだら、これを基底として、次のようにベクトルで表す。

$$\mathbf{sweet} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{green} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{furry} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

この時、この本の中の語 banana, puppy, fruit は、次のように表される。

$$\mathbf{banana} = \begin{bmatrix} 21 \\ 9 \\ 0 \end{bmatrix} \quad \mathbf{puppy} = \begin{bmatrix} 8 \\ 1 \\ 32 \end{bmatrix} \quad \mathbf{fruit} = \begin{bmatrix} 43 \\ 19 \\ 0 \end{bmatrix}$$

語の意味の分散モデルの例

別の言葉で言えば、我々は、コーパスからのデータをこれらの語を三次元のベクトル空間に埋め込むために利用したのである。

語 banana の意味は $(21, 9, 0)$ 、語 puppy の意味は $(8, 1, 32)$ 、語 fruit の意味は $(43, 19, 0)$ ということになる。



meanings of words = FVect

これでうまくいくのだ！これで、コンピュータに、語の意味の分散モデルを与えたことになる。

二つの語のベクトルの内積を計算できれば、それらのベクトルが近さがわかり、もしもそれらが同じ意味を持っていたとすれば、それを正確に表現できる。これは、NLPで仕事をしている人にはよく知られていることだ。

Coeckeらの意味論的カテゴリーは、こうして与えられる。それは、 \mathbb{R} 上の有限次元のベクトル空間である。

meanings of words = FVect

Functor: syntax \rightarrow semantics

語の分散モデルは、文には適用できない！ カテゴリー論が必要になる

残念ながら、先の語の分散モデルは、文には適用できない。文書内で同じ文が現れることはほとんどないので、先の考えではうまくいかない。

ここで、カテゴリー論が助けの手を差し出してくれる。

構成性の原理に照らせば、文の意味は、文を構成する個々の語の意味とそれらを結びつける文法のルールで計算されるべきである。

我々は、「語の意味」とその語の「文法的な型」のペアを、syntax (文法) から semantics (語の意味) への写像を通じて作ることができる。すなわち、次のような functor F を通じて。

$$F : \text{PregX} \rightarrow \text{FVect}$$

ただし、条件がある

$$F : \text{PregX} \rightarrow \text{FVect}$$

というFunctorial Semantics を利用するためには、PregXもFvectも同じカテゴリーに属していなければならない。実際、両者はともに、**compact closed categories**に属する。

Fは、**strong monoidal functor**として、compact closed category PregXの構造を、compact closed category Fvectの構造にうつす。Fによって構成性が保存されるのだ。

ただし、この重要な事実の説明は、このセッションではまだ行われていない。次回のセッションで説明する。

The Functor: Syntax \rightarrow Semantics

この節では、functor $F : \text{syntax} \rightarrow \text{semantics}$ の明示的な記述を与える。より具体的には、 $F : \text{PregX} \rightarrow \text{FVect}$ の。

オブジェクトと射についてのfunctorの働きを定義すればいい。

The Functor: Syntax \rightarrow Semantics on objects

On objects

- **F** assigns to the noun type n
a vector space $N : Fn$,
which we'll call a **noun space**
- **F** assigns to the sentence type s
a vector space $S : Fs$,
which we'll call a **sentence space**

The Functor: Syntax \rightarrow Semantics on objects

On morphisms

F assigns to a type reduction $a \xrightarrow{r} b$
a linear map $Fa \xrightarrow{Fr} Fb$

that sends

the vector corresponding to a word or phrase of
type a in Fa

to the vector corresponding to a word or phrase of
type b in Fb .

Functor $F: \text{Preg} \rightarrow \text{FVect}$

F は、compact closed structure を保存する

- F は、 $\text{Preg}\{n, s\}$ の**unit** $\eta_n^r: 1 \rightarrow n^r n$, $\eta_n^l: 1 \rightarrow n n^l$ を、
 FVect の**unit** $\eta_N: \mathbb{R} \rightarrow N \otimes N$ にうつす。

$$F(\eta_n^r) = F(\eta_n^l) = \eta_N \quad (s \text{ についても同様})$$

- F は、 $\text{Preg}\{n, s\}$ の**counit** $\epsilon_n^r: n^r n \rightarrow 1$, $\epsilon_n^l: n n^l \rightarrow 1$ を、
 FVect の**counit** $\epsilon_N: N \otimes N \rightarrow \mathbb{R}$ にうつす。

$$F(\epsilon_n^r) = F(\epsilon_n^l) = \epsilon_N \quad (s \text{ についても同様})$$

Functor $F: \text{Preg} \rightarrow \text{FVect}$

F は、compact closed structure を保存する

- F は、 $\text{Preg}\{n, s\}$ の**dual**を、 FVect の**dual**にうつす。
 $F(n^r) = F(n^l) = N^*$ ただし FVect は有限次元なので $N^* \cong N$

$$F(n^r) = F(n^l) = N \quad (s \text{ についても同様})$$

- F は、 $\text{Preg}\{n, s\}$ の**複合型**を、 FVect の**テンソル積**にうつす。

$$\text{e.g. } F(n^r s n^l) \cong F(n^r) \otimes F(s) \otimes F(n^l) = N \otimes S \otimes N$$

DisCoCat入門 2



カテゴリー論と意味の形式的理論

“bananas are fruit” の意味を計算する

Step 1: それぞれの語に、 $\text{Preg}(n, s)$ の文法型を割り当てる

banana $\rightarrow n$

are $\rightarrow n^r s n^l$

fruit $\rightarrow n$

Step 2: 「名前」空間 $N = F^n$ 「文」空間 $S = F^s$ を、
基底を選ぶことで固定する

「名前」空間 N を、次の三つのベクトルが基底 w_1, w_2, w_3 として張る三次元ベクトル空間としよう。

sweet , green , furry

$$w_1 = \text{sweet} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad w_2 = \text{green} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad w_3 = \text{furry} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

これらの基底が「名前」空間を作る

簡単にするために、 S を「真か偽か」の空間と定義して、1つのベクトル $\vec{1}$ で張られる1次元のベクトル空間とする。原点 $0 \in S$ は "偽" に対応し、 $\vec{1}$ は "真" に対応する。

$\vec{1}$ のスカラー倍はどうなるだろうか？

望むならば、 $\vec{1}$ の正のスカラー倍は、本当に真である文の意味ベクトルだと考えてもかまわない。スカラーが大きければ大きいほど、その文は正しいと考えればいい。

最後に、「名詞」空間 N と「文」空間 S が決まれば、動詞の空間はタダで手に入ることに注意しよう。

$$F(n^r sn^l) = N \otimes S \otimes N$$

Step 3: 文中の語のベクトル表現を決める

$$\mathbf{bananas} = \begin{bmatrix} 21 \\ 9 \\ 0 \end{bmatrix}, \quad \mathbf{fruit} = \begin{bmatrix} 43 \\ 19 \\ 0 \end{bmatrix}$$

これらのベクトルは、両方とも「名前」空間 N に属する。なぜなら、それらの言葉は文法型 n を持つからだ。

この語の意味のベクトル表現は、文の意味のベクトル表現を計算するのに利用される。

Step 1 で、'are' は文法型 $n^r sn^l$ を持っていることを知っている。それゆえ、それはテンソル積 $N \otimes S \otimes N$ のベクトルである。

そして、それはいいニュースである。なぜなら、もしも我々が「何々は何々である」ということを知っていれば、すなわち「子犬は怒っている」、「子犬は緑ではない」等々、その時、この 'are' のベクトルを、 3×3 の行列として再表現できる。

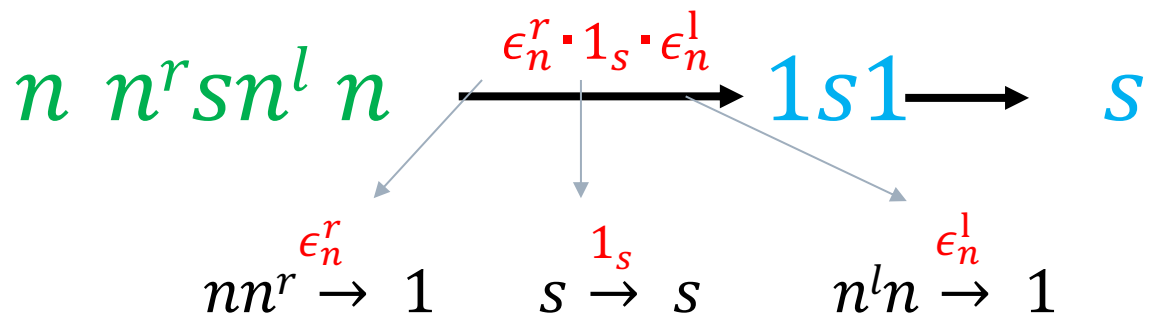
この行列 c の i 行 j 列目の係数 c_{ij} は、次のようになる。

$$c_{ij} = \begin{cases} 1, & \text{if } w_i \text{ is } w_j, \\ 0, & \text{otherwise} \end{cases}$$

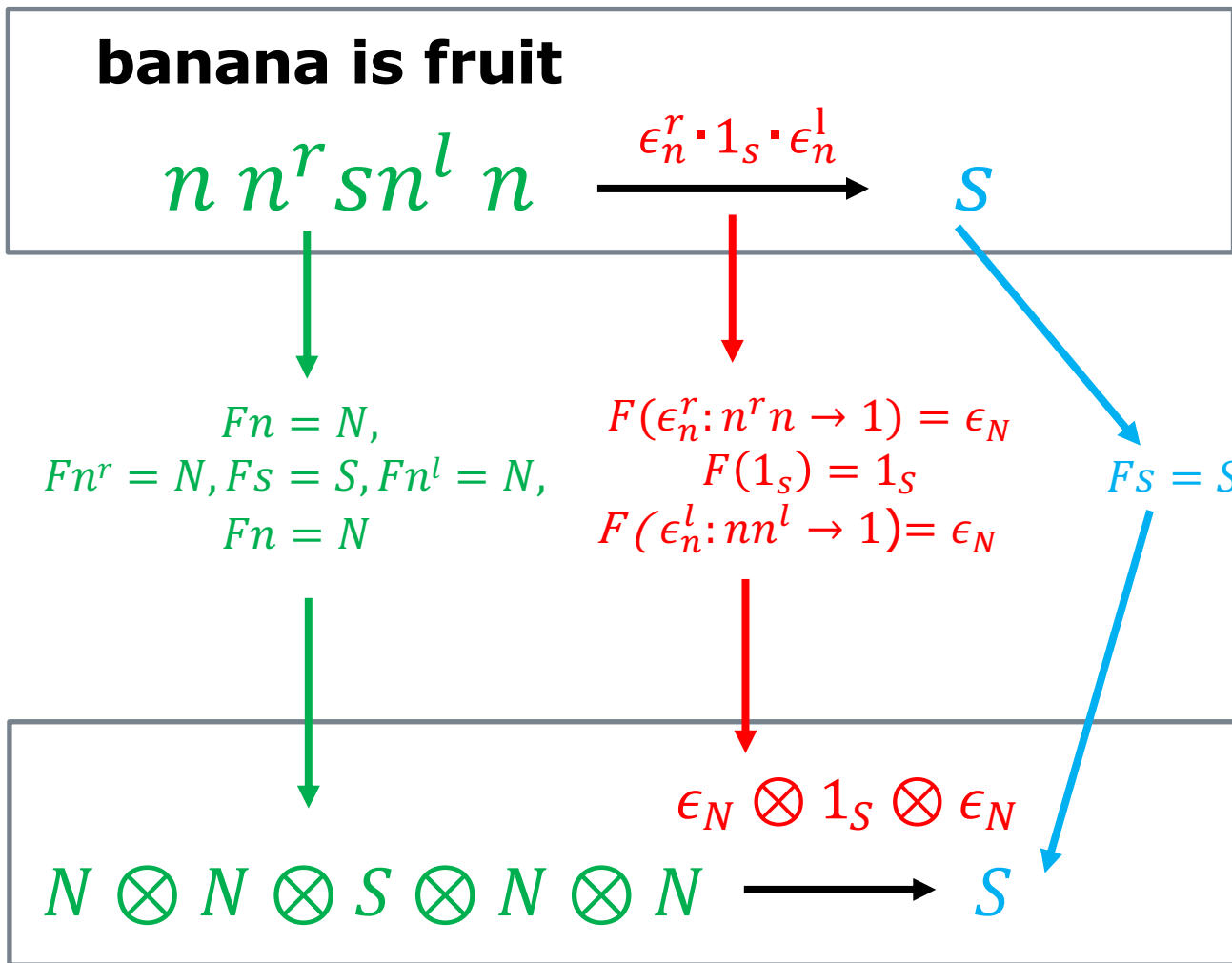
$$\mathbf{are} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 4: Preg(n, s)で、型の還元を行う

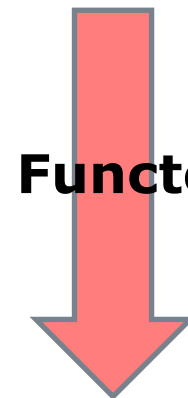
banana is fruit



Step 5: Functor F を適用する！ Syntax → Semantics



Syntax



Functor F

Semantics

Sの意味のベクトル表示は？

$$N \otimes N \otimes S \otimes N \otimes N \xrightarrow{\epsilon_N \otimes 1_S \otimes \epsilon_N} S$$

ここで、 ϵ_N は、 $\epsilon_N: N \otimes N \rightarrow \mathbb{R}$ という線型写像である。
また、 1_S は、 $1_S: S \rightarrow S$ という単位写像である。

これらの線形写像を、文に対応するベクトルに適用すると、

$$\begin{aligned} & \epsilon_N \otimes 1_S \otimes \epsilon_N(\mathbf{banana} \otimes \mathbf{is} \otimes \mathbf{fruit}) \\ &= [21 \ 9 \ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 43 \\ 19 \\ 0 \end{bmatrix} = 1974 \end{aligned}$$

文 "banana is fruit" の意味は、 $1974\vec{1}$ と表示される。

DisCoCat Coecke's original form



カテゴリー論と意味の形式的理論

Mathematical Foundations for a Compositional Distributional Model of Meaning

Bob Coecke, Mehrnoosh Sadrzadeh, Stephen Clark

2010/03/23

<https://arxiv.org/abs/1003.4394>

Around 2008 @ Oxford Uni. there were 3 people:

- One knew **grammar algebra**:

Mehrnoosh
Sadrzadeh



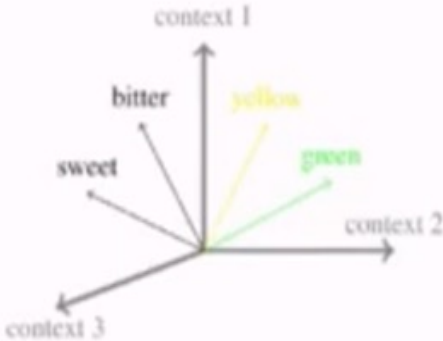
LambekのPregroupだ

$$n \cdot {}^{-1}n \cdot s \cdot n^{-1} \cdot n \leq 1 \cdot s \cdot 1 \leq s$$

“Word2Vec”

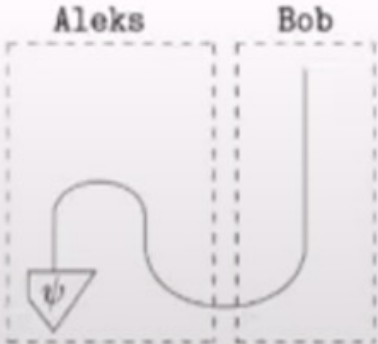
- One knew **vector-space NLP**:

Stephen Clark



- One (=me) knew **categorical QM**:

Bob Coecke





Bob Coecke



Abstract

我々は、ベクトル空間モデルによる分散意味論と、Lambekによって導入されたPregroup代数に依拠した文法的型の構成的理論を統合する数学的枠組みを提案する。

この数学的枠組みにより、適切に型付けされた文の意味を、その構成要素の語の意味から計算することができる。

具体的には、Pregroupの型の還元はカテゴリの射に「リフト」され、構成要素の語の意味を(よくタイプされた)文全体の意味に変換するプロセスとなる。

重要なのは、文全体の意味は、その文の文法構造とは独立に、ある単一の空間に存在することである。それゆえ、分散モデルでの単語の意味の比較と同じように、この空間での内積を、任意の文の意味の比較に利用することができる。

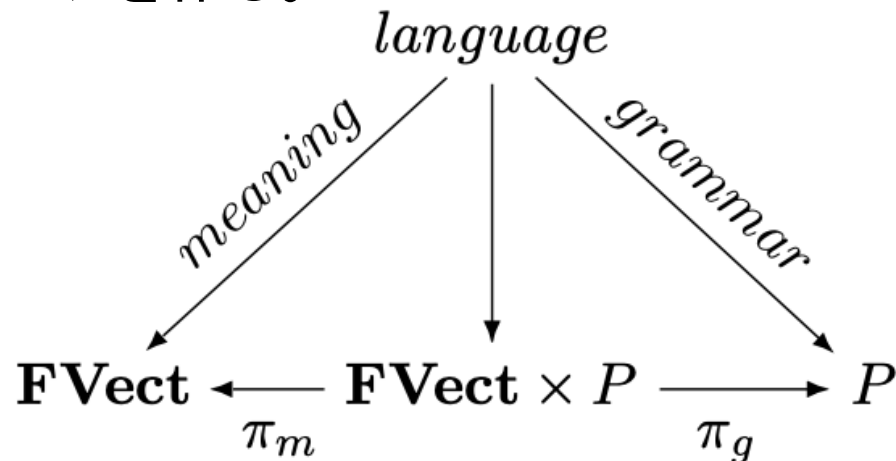
我々が用いる数学的構造は、文全体の意味を作り出すために、文の中の単語間の情報の流れを明らかにする、純粹にダイアグラムの計算を可能にする。

ベクトル空間のスカラーをブール半環に制限するような、我々の「カテゴリーモデル」のある変種は、モンタギュー式のブール値意味論を実現する。

はじめに(要旨)

意味の記号論と分散意味論は、ある意味直交する競合する理論である。それぞれに長所と短所がある、前者は構成的だが定性的で、後者は非構成的だが定量的である。認知科学の文脈でも、このところのコネクショニスト的モデルと記号論的モデルの間に類似の問題が存在する。

この論文では、テンソル空間を利用して、語の意味のベクトルと語の文法的型のペアを作る。



文法としてのPregroupが特に興味深く、我々がそれを使用する動機となったのは、カテゴリー論的な観点に立つと、それがベクトル空間やテンソル積と共通の構造を持つということである。

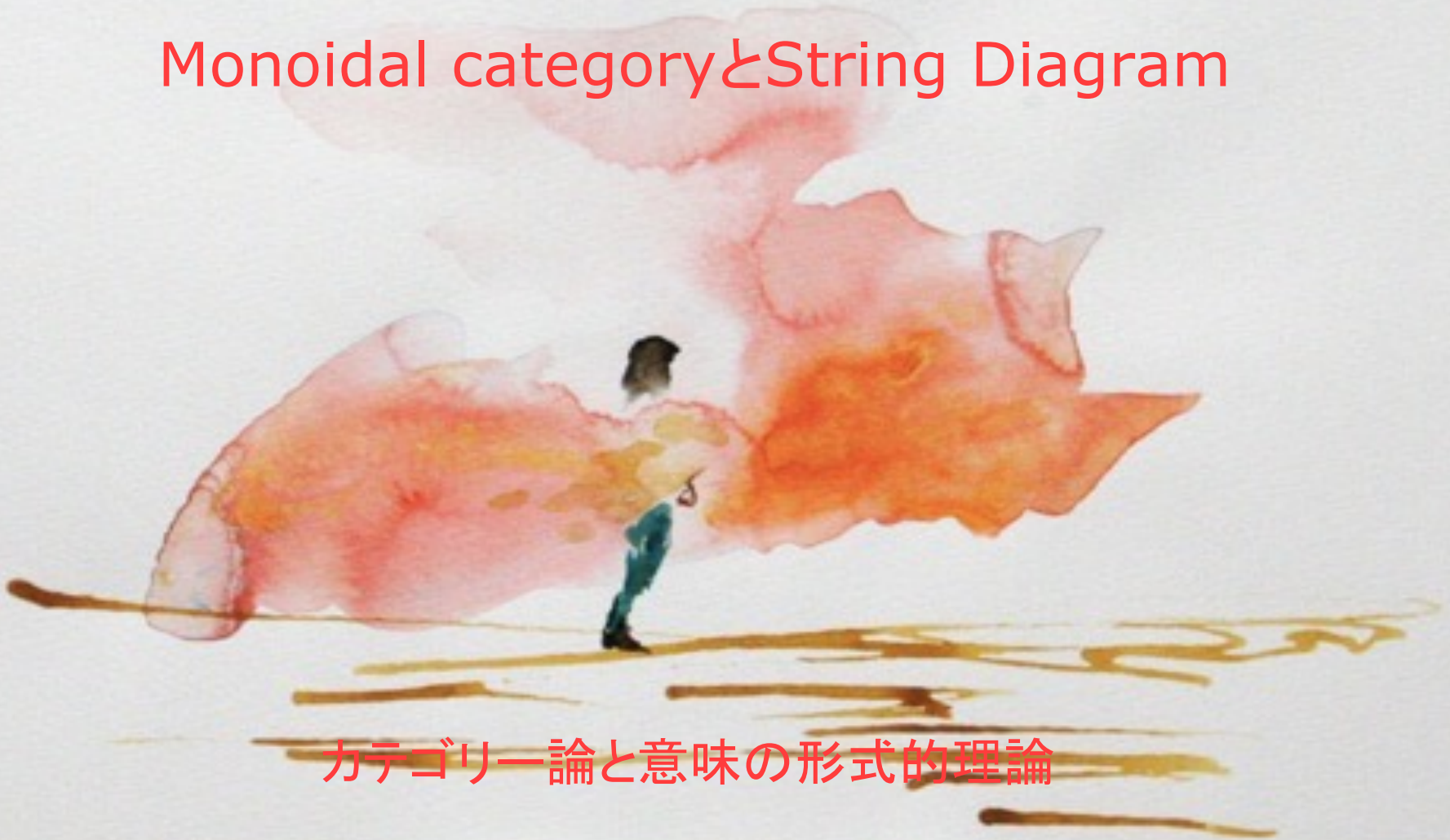
ベクトル空間、線形写像、テンソル積のカテゴリも、Pregroupも、いわゆるcompact closedカテゴリと呼ばれるものの例である。具体的には、Pregroupでの型の並置は、monoidalカテゴリのmonoidalテンソルに対応する。

文の意味を計算する数学的構造は、意味と型の二つを組み合わせたcompact closedカテゴリとなる。

言葉の意味はベクトル空間のベクトル、文法的役割はPregroupの型、(意味、型)対の合成にはPregroupでの合成と対になったベクトル空間のテンソル積が使われる。

Interlude

Monoidal categoryとString Diagram



カテゴリー論と意味の形式的理論

数式を図形で表す

Monoidal categoryとString Diagram

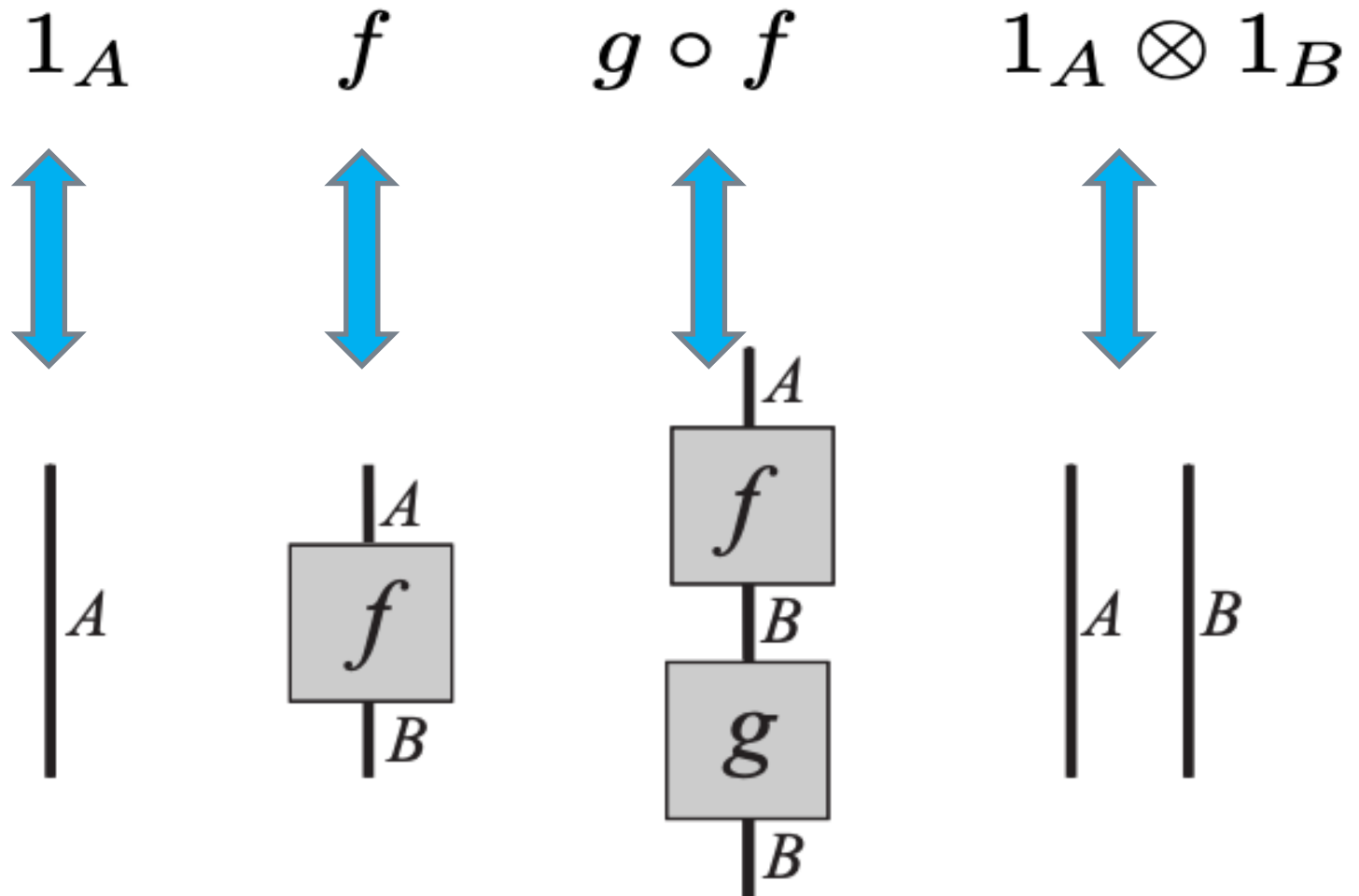
数学的証明＝図形的導出

Monoidal カテゴリーの射についての等式がmonoidal カテゴリーの公理から証明可能であれば、その等式はその図形的言語であるString Diagramでも導出可能である。

また、その逆も成り立つ。

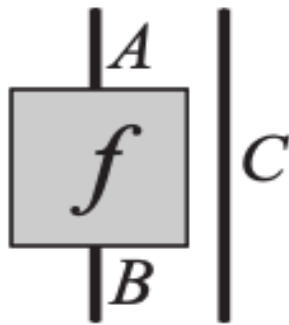
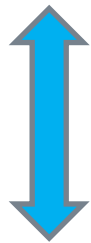
すなわち、String Diagramで図形的に導出可能な等式は、monoidal カテゴリーの公理から証明可能である。

式と図形との基本的な対応

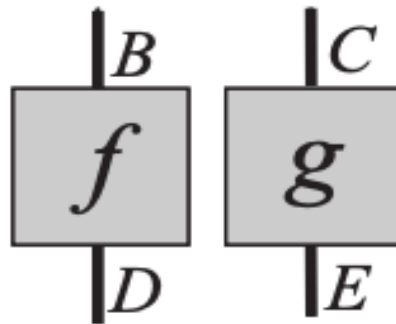
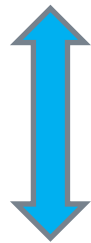


式と図形との基本的な対応

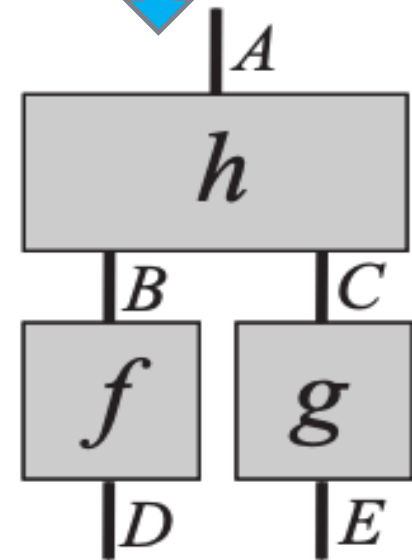
$$f \otimes 1_C$$



$$f \otimes g$$



$$(f \otimes g) \circ h$$

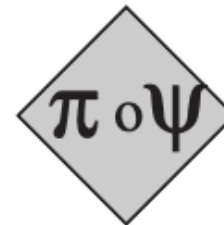
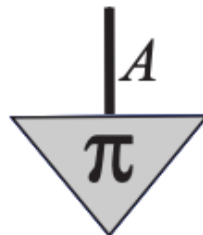
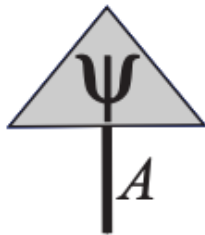
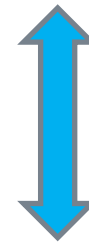
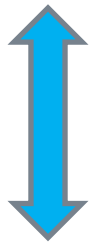
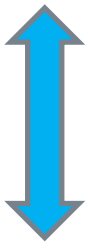


式と図形との基本的な対応

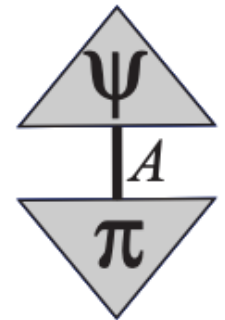
$$\psi : I \rightarrow A$$

$$\pi : A \rightarrow I$$

$$\pi \circ \psi : I \rightarrow I$$



=



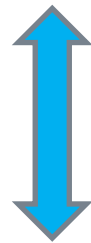
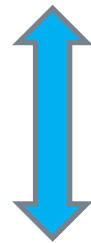
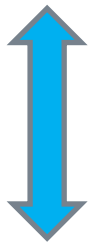
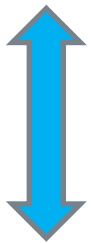
入力を持たず
出力のみを持つ
「状態」

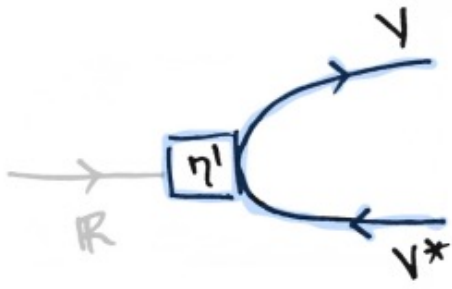
出力を持たず
入力のみを持つ
「効果・テスト」

“Picturing Quantum Processes”と逆のnotationになっている

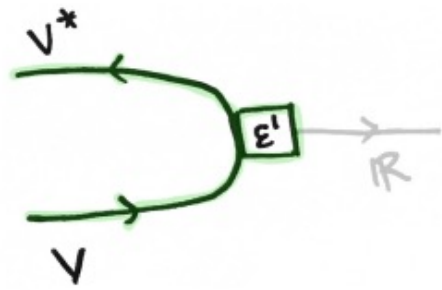
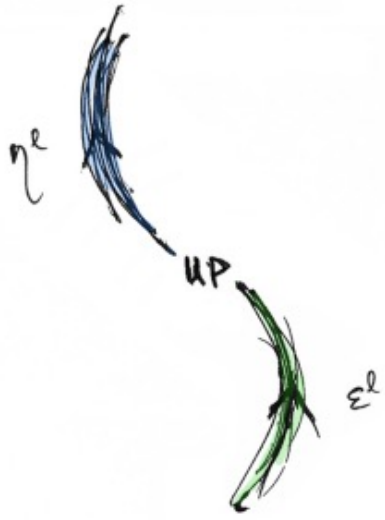
式と図形との基本的な対応

$$\eta^l : I \rightarrow A \otimes A^l \quad \epsilon^l : A^l \otimes A \rightarrow I \quad \eta^r : I \rightarrow A^r \otimes A \quad \epsilon^r : A \otimes A^r \rightarrow I$$

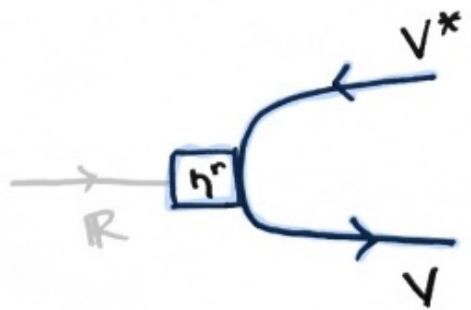




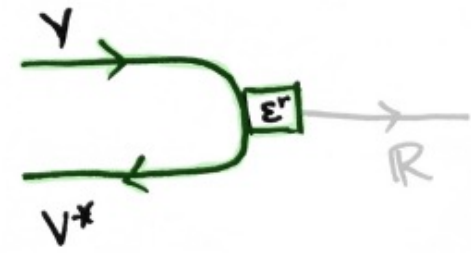
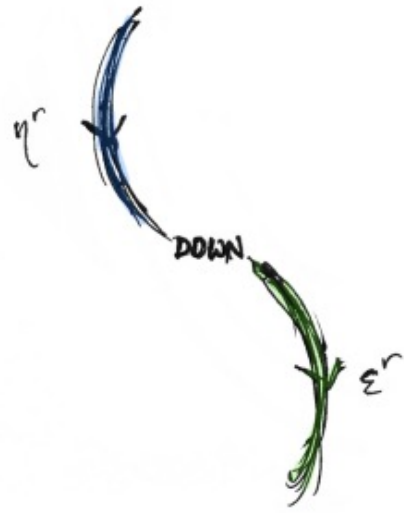
$$\mathbb{R} \xrightarrow{\eta^l} V \otimes V^*$$



$$V^* \otimes V \xrightarrow{\epsilon^l} \mathbb{R}$$



$$\mathbb{R} \xrightarrow{\eta^r} V^* \otimes V$$

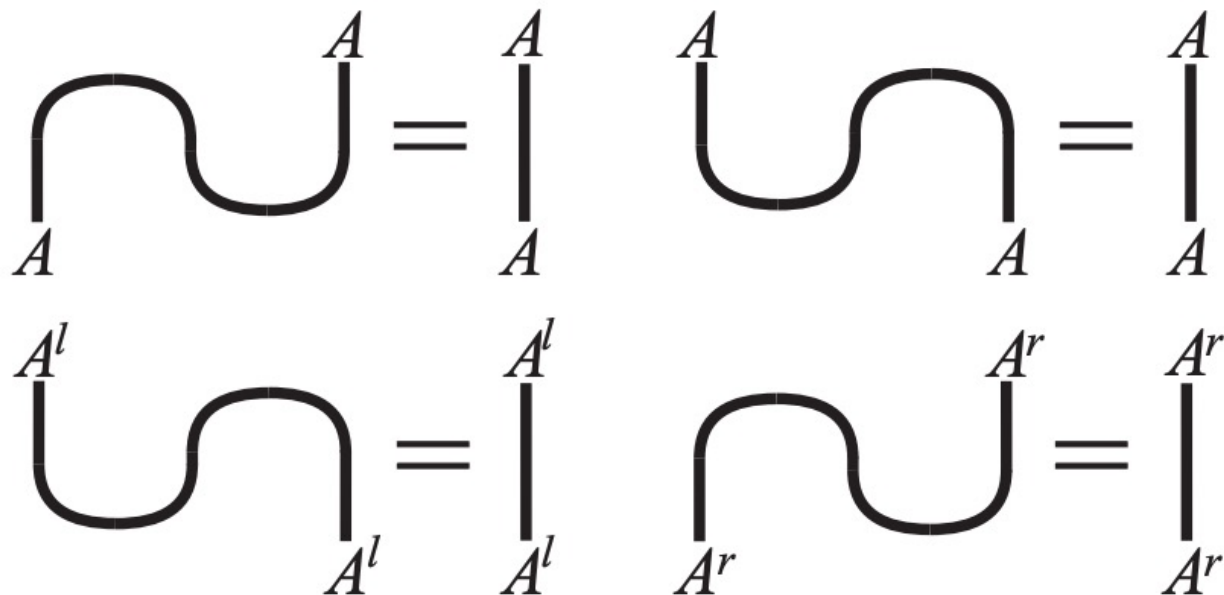


$$V \otimes V^* \xrightarrow{\epsilon^r} \mathbb{R}$$

図形を変形する等式 Yanking 等式

$$(1_A \otimes \epsilon^l) \circ (\eta^l \otimes 1_A) = 1_A \quad (\epsilon^r \otimes 1_A) \circ (1_A \otimes \eta^r) = 1_A$$

$$(\epsilon^l \otimes 1_{A^l}) \circ (1_{A^l} \otimes \eta^l) = 1_{A^l} \quad (1_{A^r} \otimes \epsilon^r) \circ (\eta^r \otimes 1_{A^r}) = 1_{A^r}$$



Yanking等式

$$(\epsilon \otimes \text{id}_V) \circ (\text{id}_V \otimes \eta) = \text{id}_V$$

$$(\text{id}_{V^*} \otimes \epsilon) \circ (\eta \otimes \text{id}_{V^*}) = \text{id}_{V^*}$$

第一の等式

第二の等式

通常の
ノテーション

$$(\epsilon \otimes \text{id}_V) \circ (\text{id}_V \otimes \eta) = \text{id}_V$$

$$(\text{id}_{V^*} \otimes \epsilon) \circ (\eta \otimes \text{id}_{V^*}) = \text{id}_{V^*}$$

$$\eta^r: R \rightarrow V \otimes V^* \quad \epsilon^r: V \otimes V^* \rightarrow R$$

$$\eta^r: R \rightarrow V^* \otimes V \quad \epsilon^r: V \otimes V^* \rightarrow R$$

Arrow

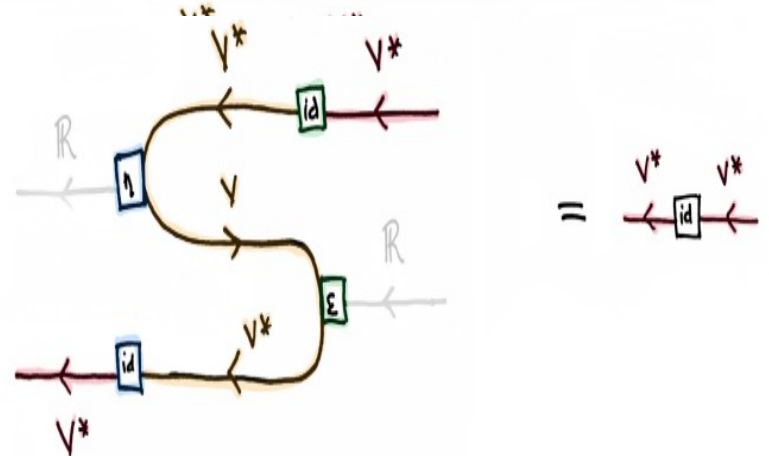
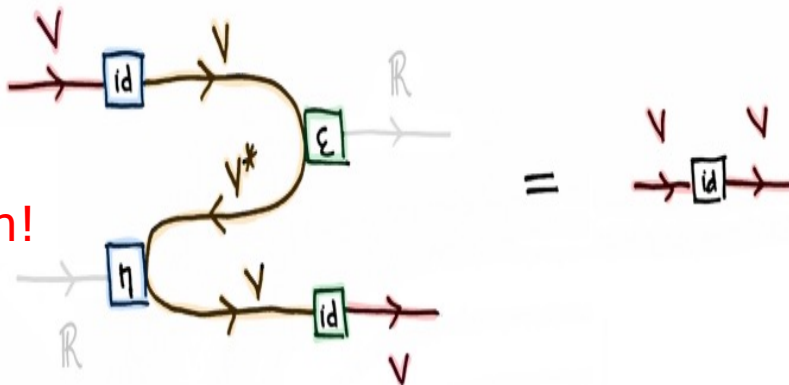
$$V \cong V \otimes R \rightarrow V \otimes V^* \otimes V \rightarrow R \otimes V \cong V$$

$$\cong V \rightarrow V$$

$$V^* \cong R \otimes V^* \rightarrow V^* \otimes V \otimes V^* \rightarrow V^* \otimes R \cong V^*$$

$$\cong V^* \rightarrow V^*$$

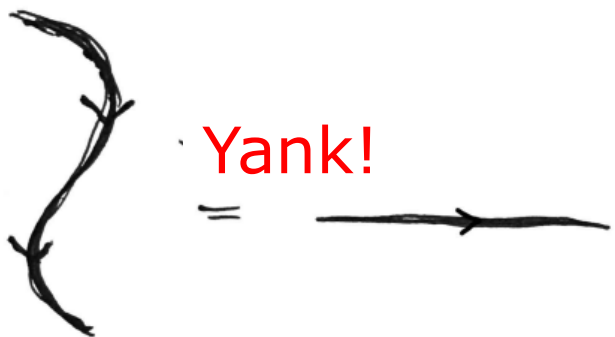
String
Diagram!



Yanking

$$(\epsilon \otimes \text{id}_V) \circ (\text{id}_V \otimes \eta) = \text{id}_V$$

$$(\text{id}_{V^*} \otimes \epsilon) \circ (\eta \otimes \text{id}_{V^*}) = \text{id}_{V^*}$$



第一の等式



第二の等式

String Diagram については 次のページを参照してほしい

<https://www.marulabo.net/docs/category01/>



String Diagram を学ぶ
-- カテゴリー論入門 (1)

DisCoCat

Coecke's diagrammatic calculus



カテゴリー論と意味の形式的理論

Mathematical Foundations for a Compositional Distributional Model of Meaning

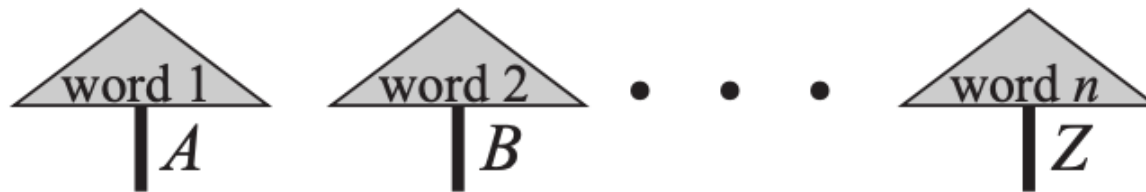
Bob Coecke, Mehrnoosh Sadrzadeh, Stephen Clark

2010/03/23

<https://arxiv.org/abs/1003.4394>

「語の意味から文の意味へ」の
プロセスを図で表す

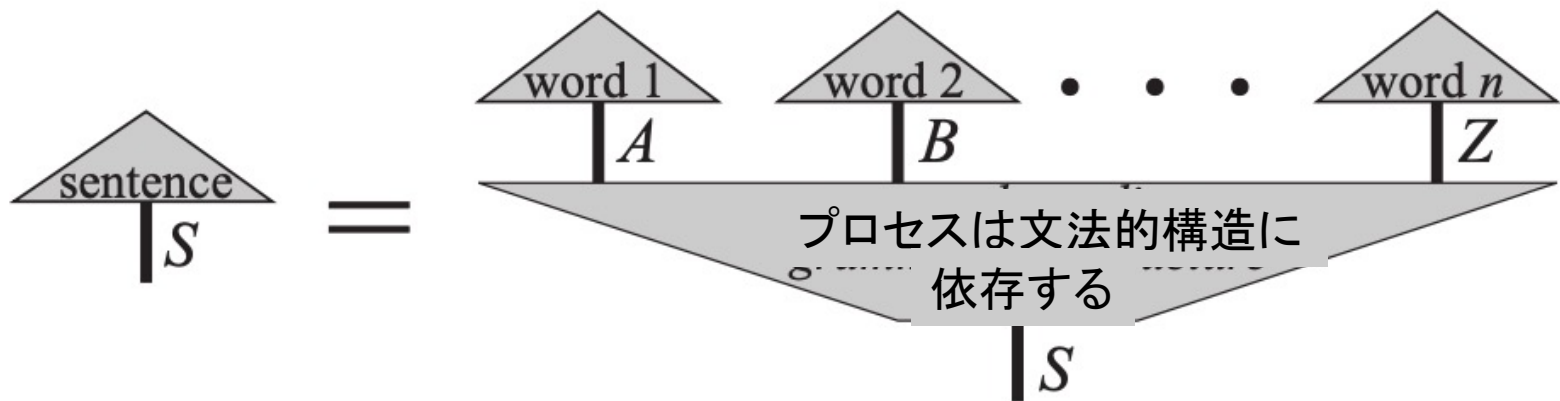
「語の意味」の並び



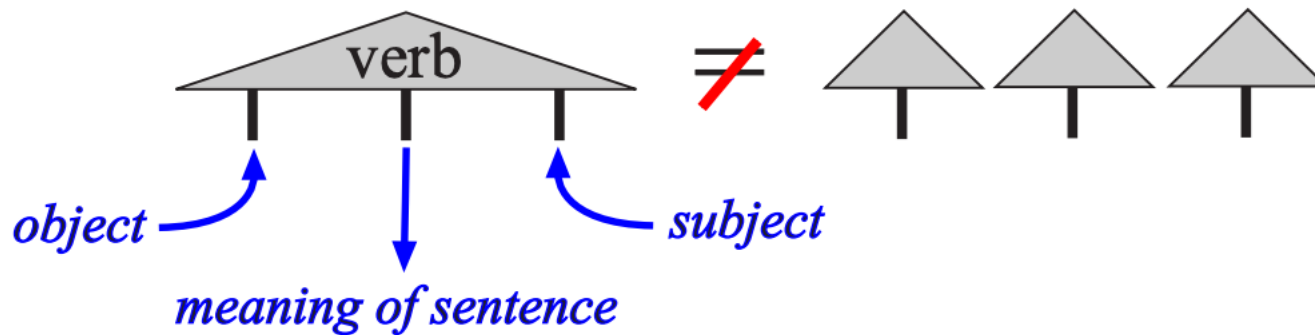
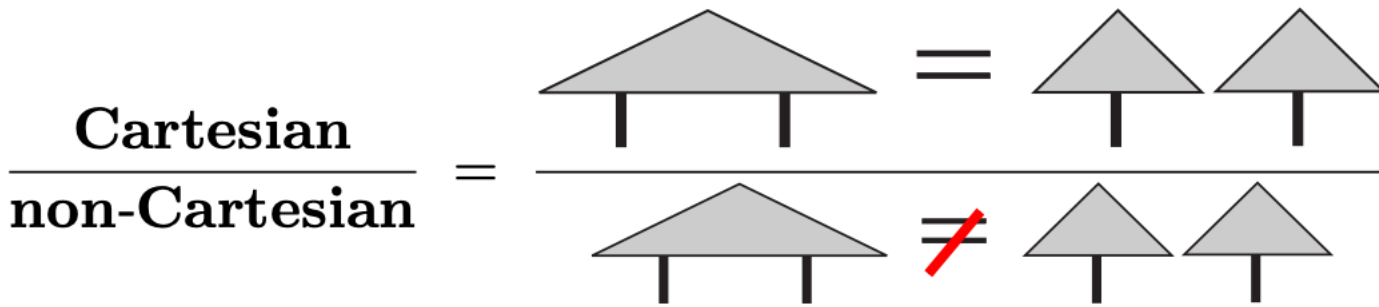
「文の意味」



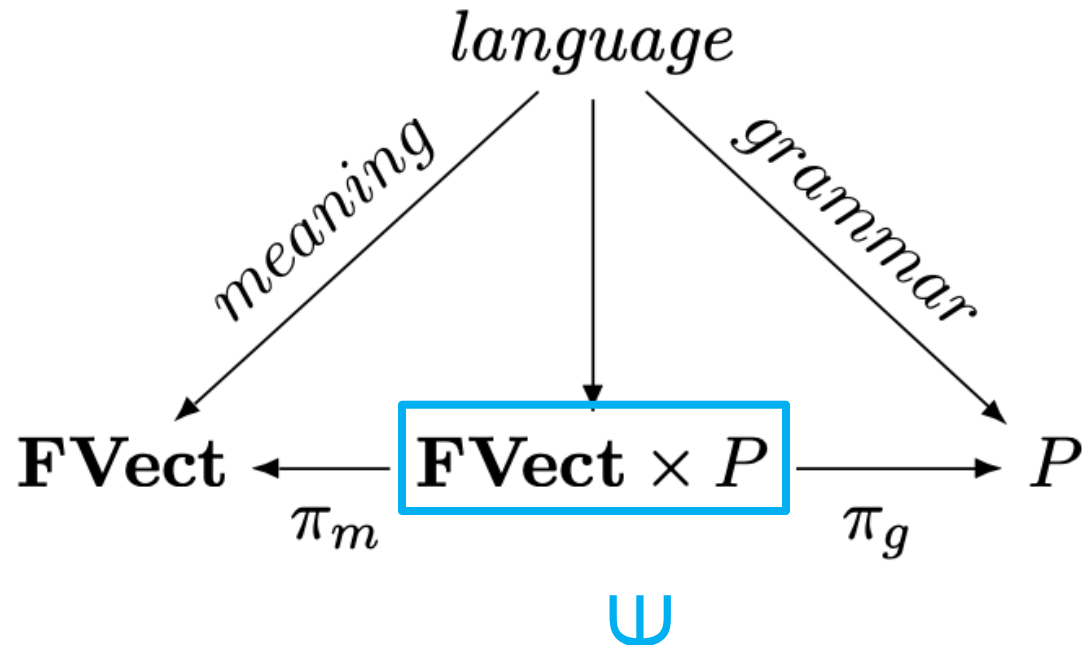
「語の意味から文の意味へ」のプロセス



「分離不能」なプロセスとしての 「文の意味」



語と文の「意味」と「文法」を
同時に表現するカテゴリー $FVect \times P$



語 w_i の意味を表す拡張された空間: (W_i, p_i)

$W_i \in FVect$: 語 w_i の意味ベクトル

$p_i \in P$: 語 w_i の文法型

語の並び $w_1 \cdots w_n$ の意味ベクトル

語の並び $w_1 \cdots w_n$ の意味ベクトルを、 $\overrightarrow{w_1 \cdots w_n}$ と表す。

この時、 $\overrightarrow{w_i} = (W_i, p_i)$ を、語 w_i の意味空間として、次のような線型写像 f を考える。

$$\overrightarrow{w_1 \cdots w_n} = f(\overrightarrow{w_1} \otimes \cdots \otimes \overrightarrow{w_n})$$

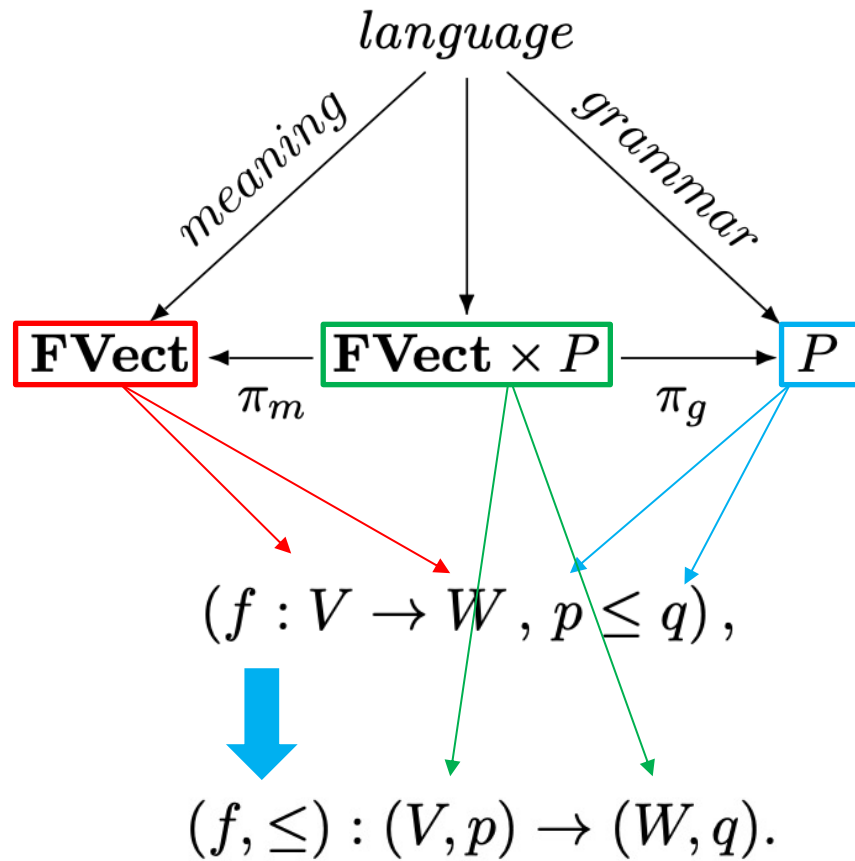
$\alpha = [p_1 \cdots p_n \leq x]$ なる型 x の文法的導出が存在する時、線型写像 f は、 α 中の p_i を W_i に置き換えることで作ることができる。

FVect × P 内の射としての「文の意味」

$\alpha = [p_1 \cdots p_n \leq x]$ は \mathbf{P} の射であり、 $f = \alpha[p_i \setminus W_i]$ は、**FVect** の線型写像なので、次のものは、**FVect** × \mathbf{P} の射である。

$$(W_1 \otimes \cdots \otimes W_n, p_1 \cdots p_n) \xrightarrow{(f, \leq)} (X, x)$$

この f を、「語の意味から文の意味への写像」と呼ぶ。



$$(\eta^l, \leq) : (\mathbb{R}, 1) \rightarrow (V \otimes V, p \cdot p^l)$$

$$(\eta^r, \leq) : (\mathbb{R}, 1) \rightarrow (V \otimes V, p^r \cdot p)$$

$$(\epsilon^l, \leq) : (V \otimes V, p^l \cdot p) \rightarrow (\mathbb{R}, 1)$$

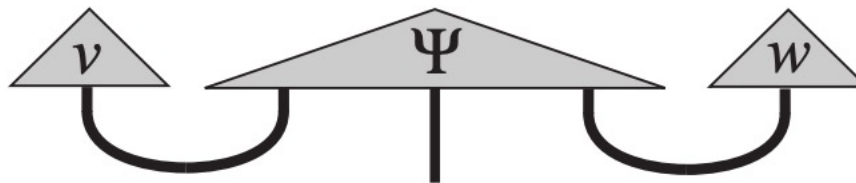
$$(\epsilon^r, \leq) : (V \otimes V, p \cdot p^r) \rightarrow (\mathbb{R}, 1)$$

“John likes Mary” の意味

John likes Mary
 n $(n^r s n^l)$ n

$n(n^r s n^l)n \rightarrow 1s n^l n \rightarrow 1s1 \rightarrow s$

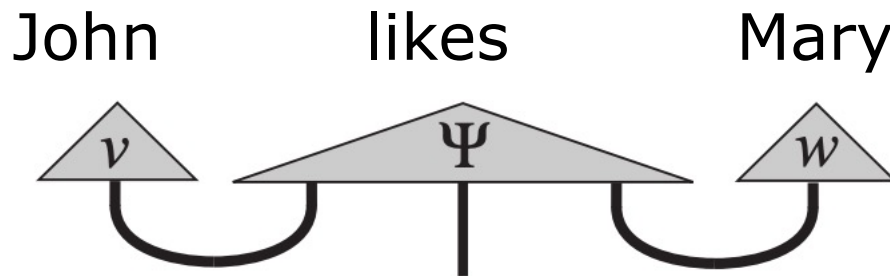
n n^r s n^l n



“John likes Mary” の意味を 図形で表現する

$$(V \otimes T \otimes W, n(n^r s n^l) n) \xrightarrow{(f, \leq)} (S, s)$$

$$f = \epsilon_V \otimes 1_S \otimes \epsilon_W : V \otimes (V \otimes S \otimes W) \otimes W \rightarrow S$$



“John likes Mary” の意味を計算する

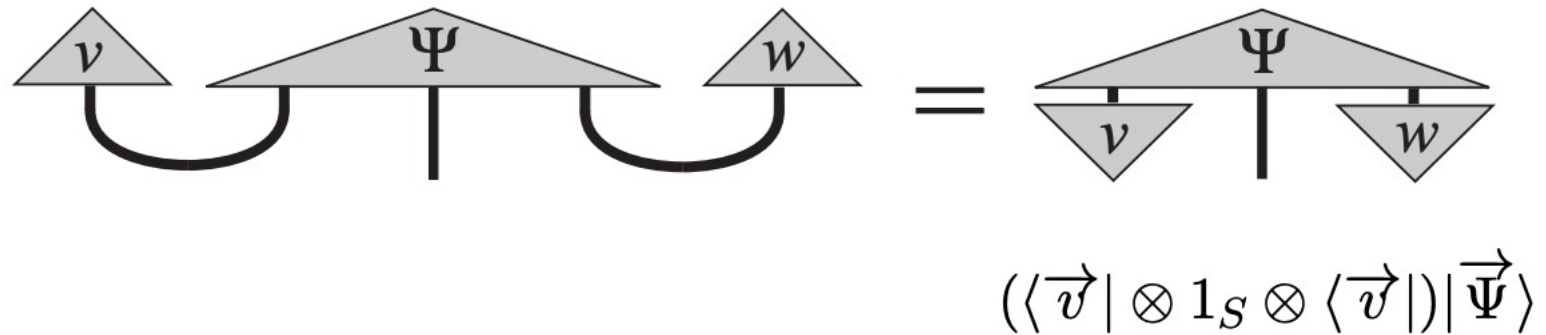
$$f = \epsilon_V \otimes 1_S \otimes \epsilon_W : V \otimes (V \otimes S \otimes W) \otimes W \rightarrow S$$

$$\Psi = \sum_{ijk} c_{ijk} \vec{v}_i \otimes \vec{s}_j \otimes \vec{w}_k \in V \otimes S \otimes W$$

$$\begin{aligned}
 f(\vec{v} \otimes \vec{\Psi} \otimes \vec{w}) &= \epsilon_V \otimes 1_S \otimes \epsilon_W(\vec{v} \otimes \vec{\Psi} \otimes \vec{w}) \\
 &= \sum_{ijk} c_{ijk} \langle \vec{v} | \vec{v}_i \rangle \vec{s}_j \langle \vec{w}_k | \vec{w} \rangle \\
 &= \sum_j \left(\sum_{ik} c_{ijk} \langle \vec{v} | \vec{v}_i \rangle \langle \vec{w}_k | \vec{w} \rangle \right) \vec{s}_j .
 \end{aligned}$$

“John likes Mary” の図形を変形する

$$(\langle \epsilon_V^r | \otimes 1_S \otimes \langle \epsilon_V^r |) | \vec{v} \rangle \otimes | \vec{\Psi} \rangle \otimes | \vec{w} \rangle$$



“John does not like Mary” の意味

John	does	not	like	Mary
n	$(n^r s j^l \sigma)$	$(\sigma^r j j^l \sigma)$	$(\sigma^r j n^l)$	n

n : noun

s : declarative statement

j : infinitive of the verb

σ : glueing type

$$n (n^r s j^l \sigma) (\sigma^r j j^l \sigma) (\sigma^r j n^l) n \rightarrow s$$

$n \quad n^r s j^l \sigma \quad \sigma^r j j^l \sigma \quad \sigma^r j n^l \quad n$

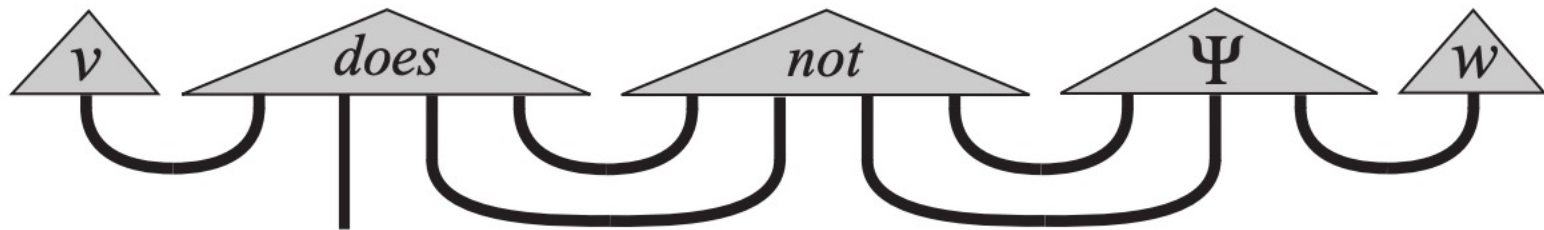


“John does not like Mary” の意味を 図形で表現する

$$f = (1_S \otimes \epsilon_J \otimes \epsilon_J) \circ (\epsilon_V \otimes 1_S \otimes 1_{J^*} \otimes \epsilon_V \otimes 1_J \otimes 1_{J^*} \otimes \epsilon_V \otimes 1_J \otimes \epsilon_W) : \\ V \otimes (V^* \otimes S \otimes J^* \otimes V) \otimes (V^* \otimes J \otimes J^* \otimes V) \otimes (V^* \otimes J \otimes W^*) \otimes W \rightarrow S$$



$$f(\vec{v} \otimes \vec{does} \otimes \vec{not} \otimes \vec{\Psi} \otimes \vec{w})$$



“does” を表す図形

$$S = J \quad \text{and} \quad \overrightarrow{\text{does}} = \sum_{ij} \vec{e}_i \otimes \vec{e}_j \otimes \vec{e}_j \otimes \vec{e}_i \in V \otimes \boxed{J \otimes J} \otimes V.$$

$$\overrightarrow{\text{does}} \simeq (1_V \otimes \eta_J \otimes 1_V) \circ \eta_V : \mathbb{R} \rightarrow V \otimes J \otimes J \otimes V$$

ここでは、射 $V \rightarrow W$ は、射 $\mathbb{R} \rightarrow V^* \otimes W$ とみなせることを使った。

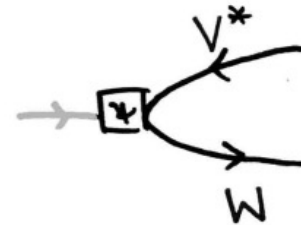
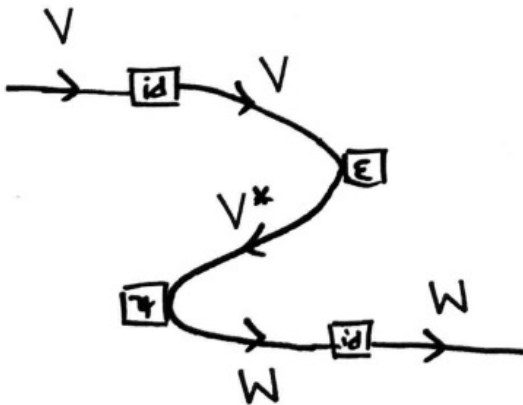
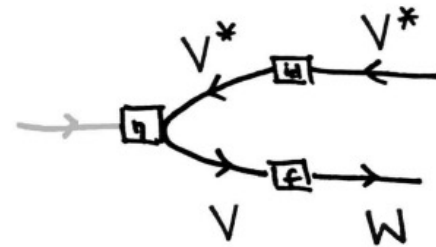
$$\text{hom}(V, W) \cong \text{hom}(\mathbb{R}, V^* \otimes W)$$

$$\text{hom}(V, W) \cong \text{hom}(\mathbb{R}, V^* \otimes W)$$

$\text{hom}(V, W)$

\cong

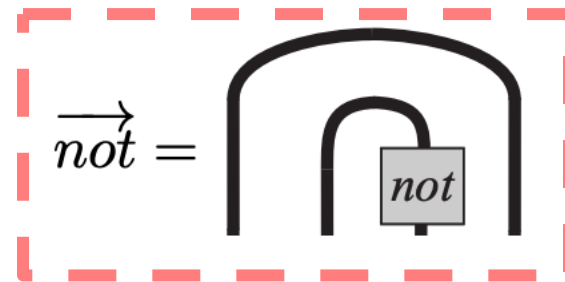
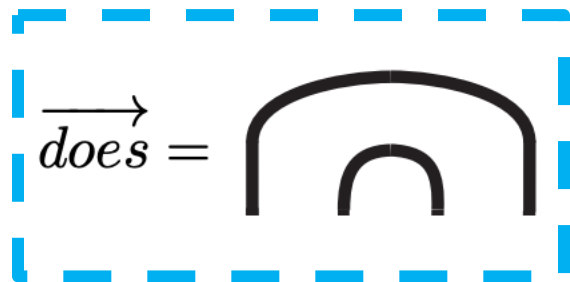
$\text{hom}(\mathbb{R}, V^* \otimes W)$



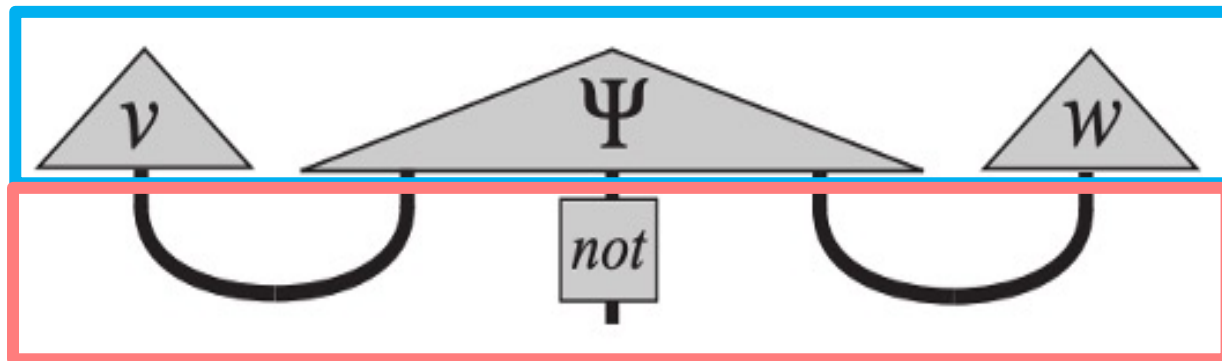
“does”と“not”を表す図形

$$\vec{does} = \sum_i \vec{e}_i \otimes (|00\rangle + |11\rangle) \otimes \vec{e}_i \in V \otimes J \otimes J \otimes V$$

$$\vec{not} = \sum_i \vec{e}_i \otimes (|01\rangle + |10\rangle) \otimes \vec{e}_i \in V \otimes J \otimes J \otimes V.$$

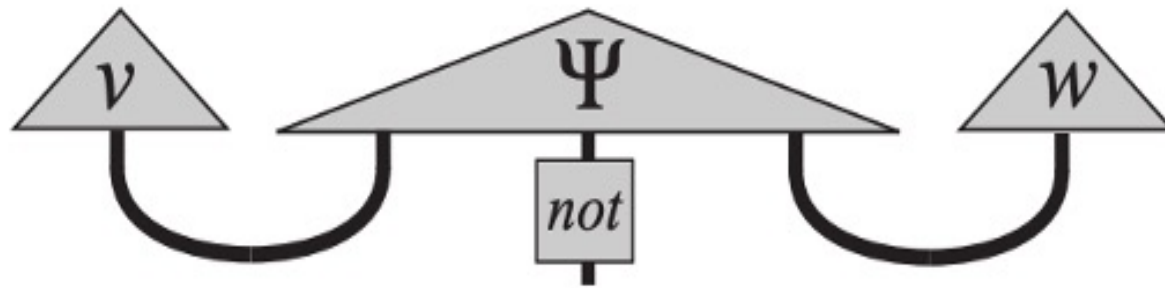


これは、次のように解釈できる



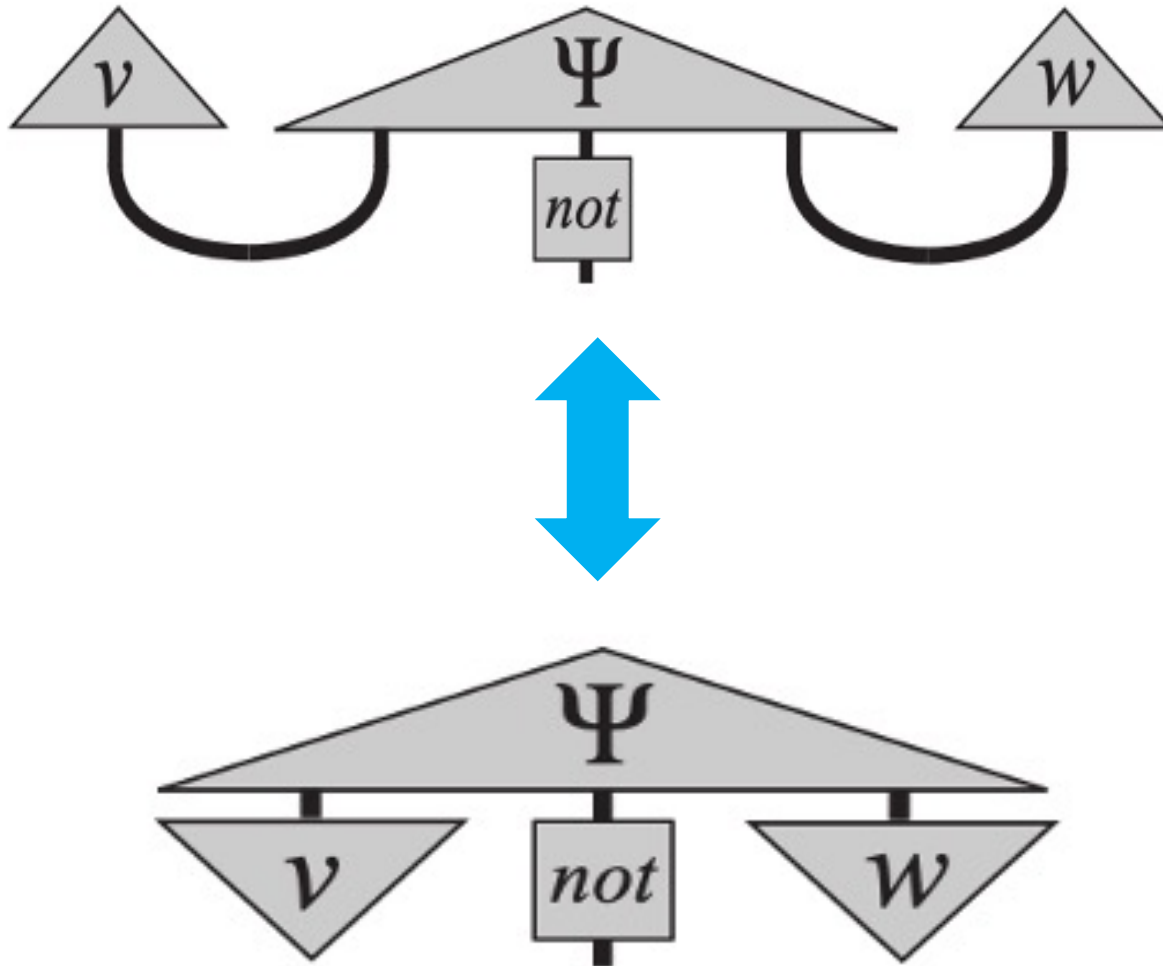
$$\left(\epsilon_V \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \epsilon_W \right) (\vec{v} \otimes \vec{\Psi} \otimes \vec{w})$$

これは、次のように解釈できる

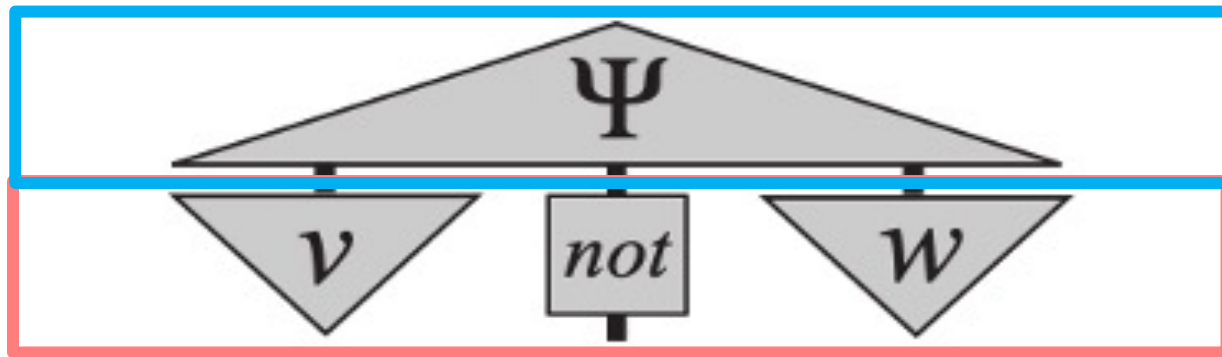


$$\left(\epsilon_V \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \epsilon_W \right) (\vec{v} \otimes \vec{\Psi} \otimes \vec{w})$$

これはまた、次のように変形できる



これは、次のように計算できる



$$(\langle \vec{v} | \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \langle \vec{w} |) |\Psi\rangle$$

“John does not like Mary” の意味を計算する

$$\begin{aligned}
 & f \left(\vec{m}_3 \otimes \left(\sum_l \vec{m}_l \otimes \bar{d} \otimes \vec{m}_l \right) \otimes \left(\sum_k \vec{m}_k \otimes \bar{n} \otimes \vec{m}_k \right) \otimes \left(\sum_{ij} \vec{m}_i \otimes \overrightarrow{like}_{ij} \otimes \vec{f}_j \right) \otimes \vec{f}_4 \right) \\
 &= h \left(\sum_{ijkl} \langle \vec{m}_3 | \vec{m}_l \rangle \bar{d} \langle \vec{m}_l | \vec{m}_k \rangle \bar{n} \langle \vec{m}_k | \vec{m}_i \rangle \overrightarrow{like}_{ij} \langle \vec{f}_j | \vec{f}_4 \rangle \right) \\
 &= h \left(\sum_{ijkl} \delta_{3l} \bar{d} \delta_{lk} \bar{n} \delta_{ki} \overrightarrow{like}_{ij} \delta_{j4} \right) \\
 &= h \left(\bar{d} \otimes \bar{n} \otimes \overrightarrow{like}_{34} \right) \\
 &= h \left((|00\rangle + |11\rangle) \otimes (|10\rangle + |01\rangle) \otimes \overrightarrow{like}_{34} \right) \\
 &= h \left(|0010\overrightarrow{like}_{34}\rangle + |0001\overrightarrow{like}_{34}\rangle + |1110\overrightarrow{like}_{34}\rangle + |1101\overrightarrow{like}_{34}\rangle \right) \\
 &= |0\rangle\langle 0 | 1\rangle\langle 0 | \overrightarrow{like}_{34}\rangle + |0\rangle\langle 0 | 0\rangle\langle 1 | \overrightarrow{like}_{34}\rangle + \\
 &\quad |1\rangle\langle 1 | 1\rangle\langle 0 | \overrightarrow{like}_{34}\rangle + |1\rangle\langle 1 | 0\rangle\langle 1 | \overrightarrow{like}_{34}\rangle \\
 &= |0\rangle\langle 1 | \overrightarrow{like}_{34}\rangle + |1\rangle\langle 0 | \overrightarrow{like}_{34}\rangle \\
 &= \begin{cases} |1\rangle & \overrightarrow{like}_{34} = |0\rangle \\ |0\rangle & \overrightarrow{like}_{34} = |1\rangle \end{cases}
 \end{aligned}$$



