
ことばと意味の数学的構造

はじめに

人間とことばで対話することができるChatGPTの登場は、「ことばと意味」についての私たちの関心を改めて高めました。

ChatGPTを産み出した「大規模言語モデル」の成功は、意味を多次元のベクトルで表現する「意味の分散表現」技術の採用に多くをおっています。

AI技術の分野で、「意味の分散表現論」がどのように生まれ、どのように発展してきたかを、理論と実装の二つの面から、振り返ってみた考察を、「意味の分散表現論の系譜 – 大規模言語モデルへ」にまとめました。ぜひ、ご利用ください。

意味の分散表現論の系譜 – 大規模言語モデルへ

- 2003年 Bengioの「次元の呪い」と語の特徴の分散表現
- 2006年 HintonのAuto Encoder 意味的ハッシング
- 2011年 RNNによる文の生成
- 2013年 Word2Vec 語の意味ベクトル
- 2014年 Sequence to Sequence 文の意味ベクトル
- 2015年 RNNの不思議な力 RNNは文法を理解している
- 2016年 Attention Mechanism
- 2016年 Google ニューラル機械翻訳
- 2017年 Transformer
- 2019年 BERT
- 2020年 GPT-3
- 2022年 ChatGPT

genealogy.pdf
https://drive.google.com/file/d/1NTs7r-wdtG1EbIkpb1oDmmIFiTlU_0ip

はじめに

小論は、同時に公開したこの「意味の分散表現論の系譜 – 大規模言語モデルへ」とは異なる視点から、「ことばと意味」について、その背後に存在する数学的構造を探究しようとする研究を紹介したものです。

なぜなら、この「ことばと意味」の形式的・数学的理論の分野でも、近年、目覚ましい研究の進展が見られるからです。

ここでもその関心は、「意味の分散表現」に置かれています。「意味の分散表現」の担い手を多次元ベクトルから密度行列に置き換えるという研究の中で、「ことばと意味」の理論の量子論との一致が発見されています。これは驚くべきことです。また、自然言語処理を量子コンピュータ上で行おうという実験が始まっています。

はじめに

残念ながら、今回のセミナーでは、こうした新しい研究の進展 **(Part 4)** については詳しく述べることはできませんでした。改めて別のセミナーで紹介したいと考えています。

今回のセミナーの中心は、AIの分野での「意味の分散表現論」とは異なる、いわば「もう一つの意味の表現論」の出発点となった、Bob Coecke, Tai-Danae Bradley らの、「カテゴリー論的構成的分散意味論」 DisCoCat 理論です。

「カテゴリー論的構成的分散意味論」を扱った**Part 3** では、オリジナルのCoeckeの展開と、それを紹介したTai-Danaeの議論の二つを紹介しました。内容は重複しますが、二人のアプローチを知ることは、DiCoCatの理解に役立つと思います。

はじめに

ことばと意味への関心は、なにも最新の大規模言語モデルによってはじめて生まれたわけではありません。

意味の数学的理論の基礎には、LawvereのFunctorial Semantics があり**(Part 1)**、ことばの構成性の理論の基礎には、Chomsky - Lambekの文法の形式化・数学化の取り組みがあります**(Part 2)**。最新の「意味の分散表現論」は、1950-60年代に起源を持つこれらの理論によって支えられています。

セミナーの最後の方で、Coeckeが、現在(2022-2023年)Oxfordで学生に教えている講義のシラバスを紹介しておきました。若い人が（若い人に限りませんが）、今、何を学ぶべきかぜひ、参考にしてもらえればと思います。

Part 1

意味の形式的理論
Functorial Semantics

- 「理論」と「モデル」
- Functorial Semantics

Part 2

ことばの構成性
Pregroup Grammar

- ChomskyとLambek
- Combinatory Categorial Grammar
- Pregroup Grammar

Part 3-1

カテゴリー論的構成的分散意味論

DisCoCatの登場

Bob Coecke

- Monoidal categoryとString Diagram
- 語の意味から文の意味へ」のプロセスを図形で表す
- 「語の意味から文の意味へ」の写像

Part 3-2

カテゴリー論的構成的分散意味論

カテゴリー論の応用としてのDisCoCat

Tai-Danae Bradley

- カテゴリー論と自然言語処理
- Syntax Category: Pregroup Grammar
- Semantics Category: Vector Space
- Functor: Syntax → Semantics
- “bananas are fruit” の意味を計算する

Part 4

カテゴリー論的構成的分散意味論の展開

- Coeckeの Quantum-NLP
- Tai-Danae のReduced Density

Part 1

意味の形式的理論
Functorial Semantics

Part 1

意味の形式的理論
Functorial Semantics

- 「理論」と「モデル」
- Functorial Semantics

「理論」と「モデル」

「理論」と「モデル」

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　→　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

形式的なルールが
与えられている。
その下で

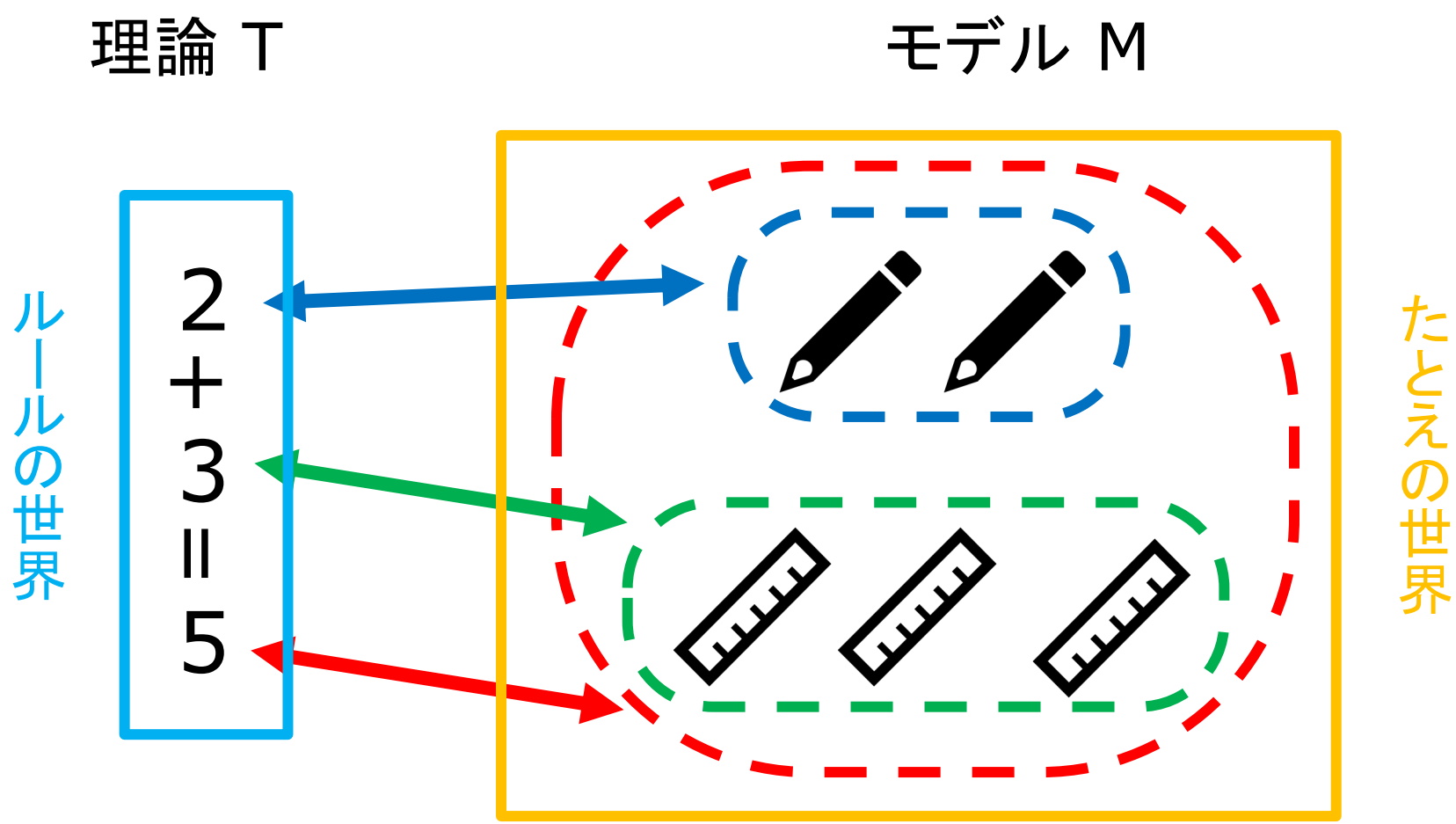
命題φは証明可能

モデル M

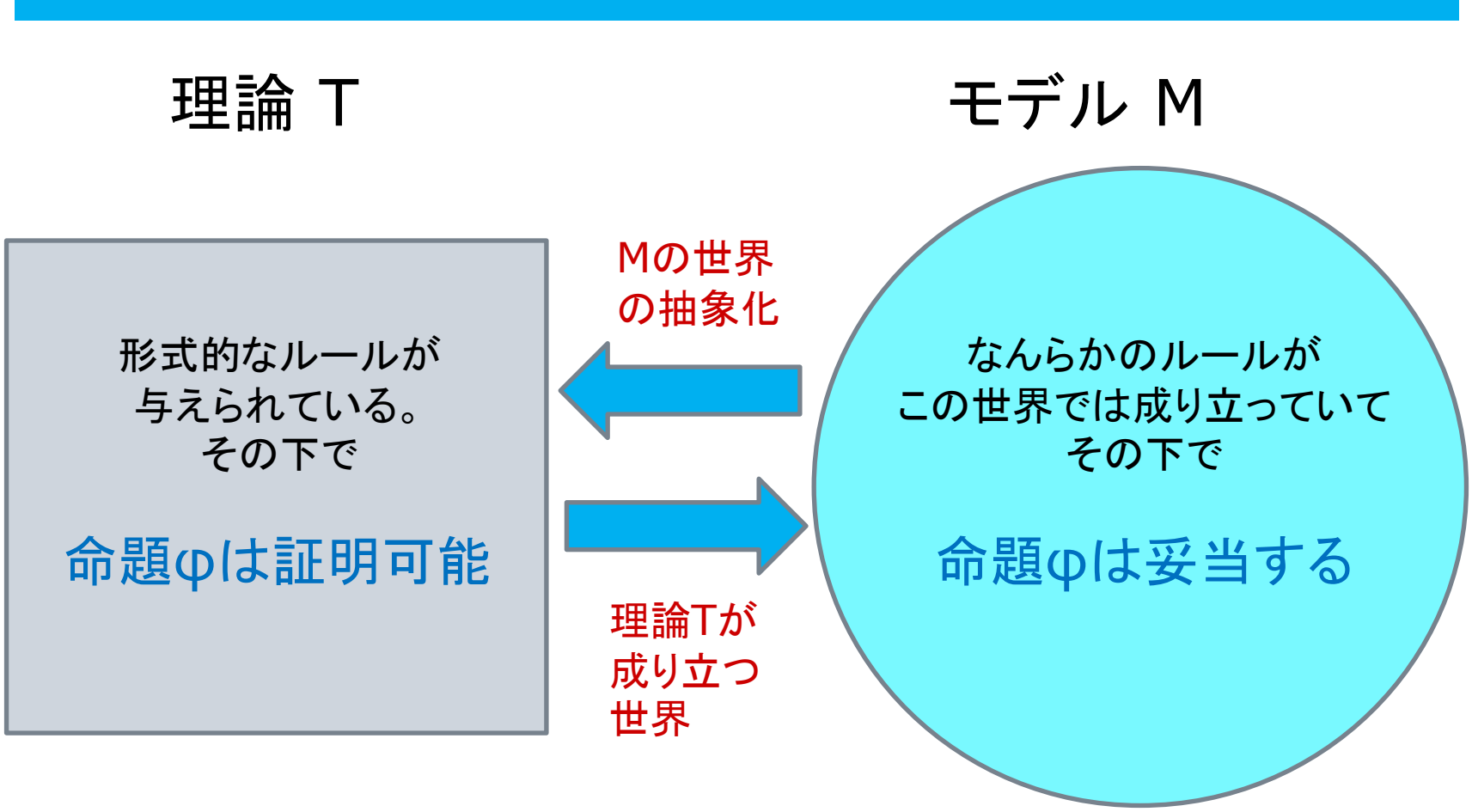
なんらかのルールが
この世界では成り立っていて
その下で

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

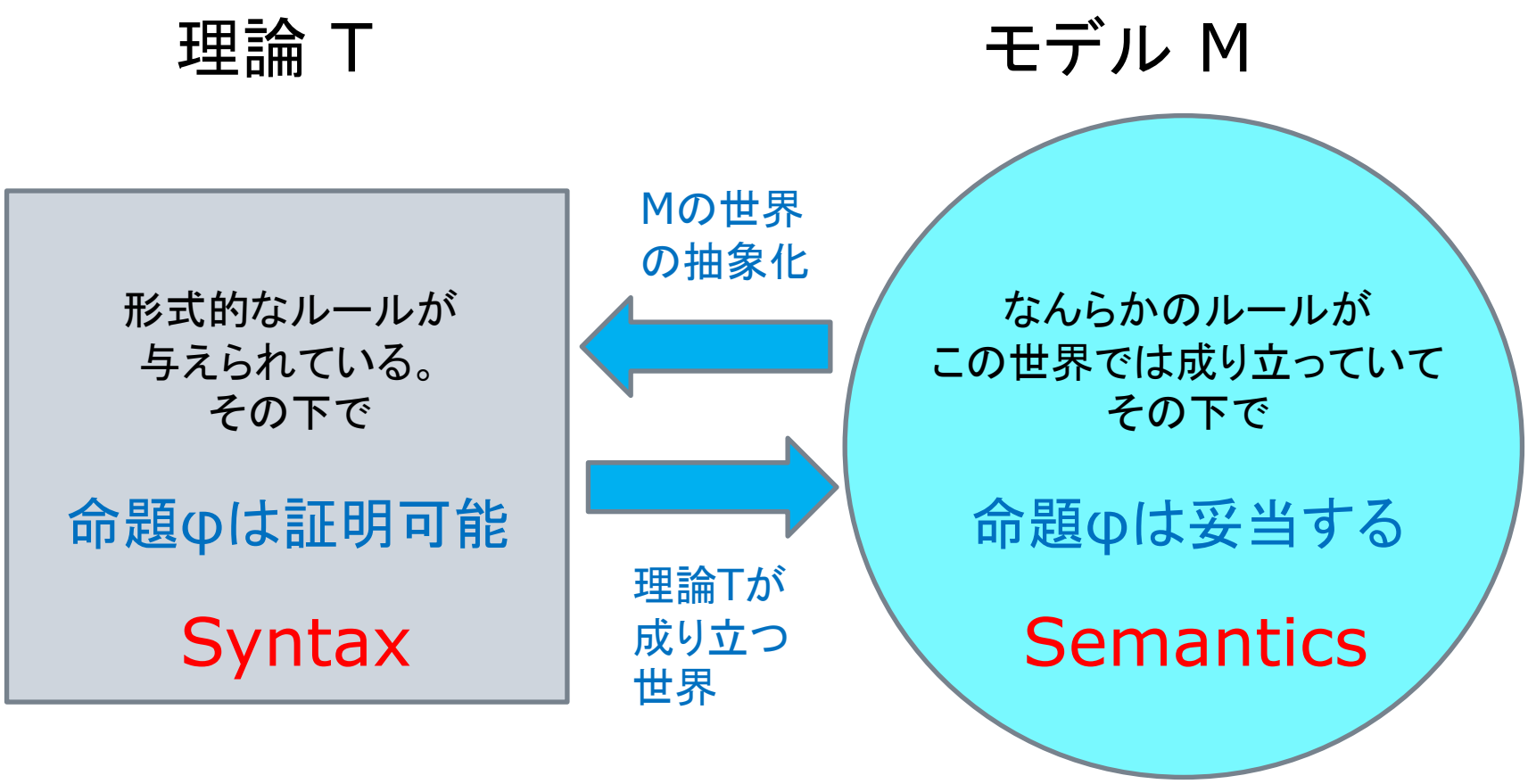


「理論」の世界と「モデル」の世界



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



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、次のような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。

「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」

この定理から、次のことが導かれる。

「ある理論が無矛盾ならそれはモデルを持つ」

https://goo.gl/5HLMsK

理論とモデルの例
群の表現

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。

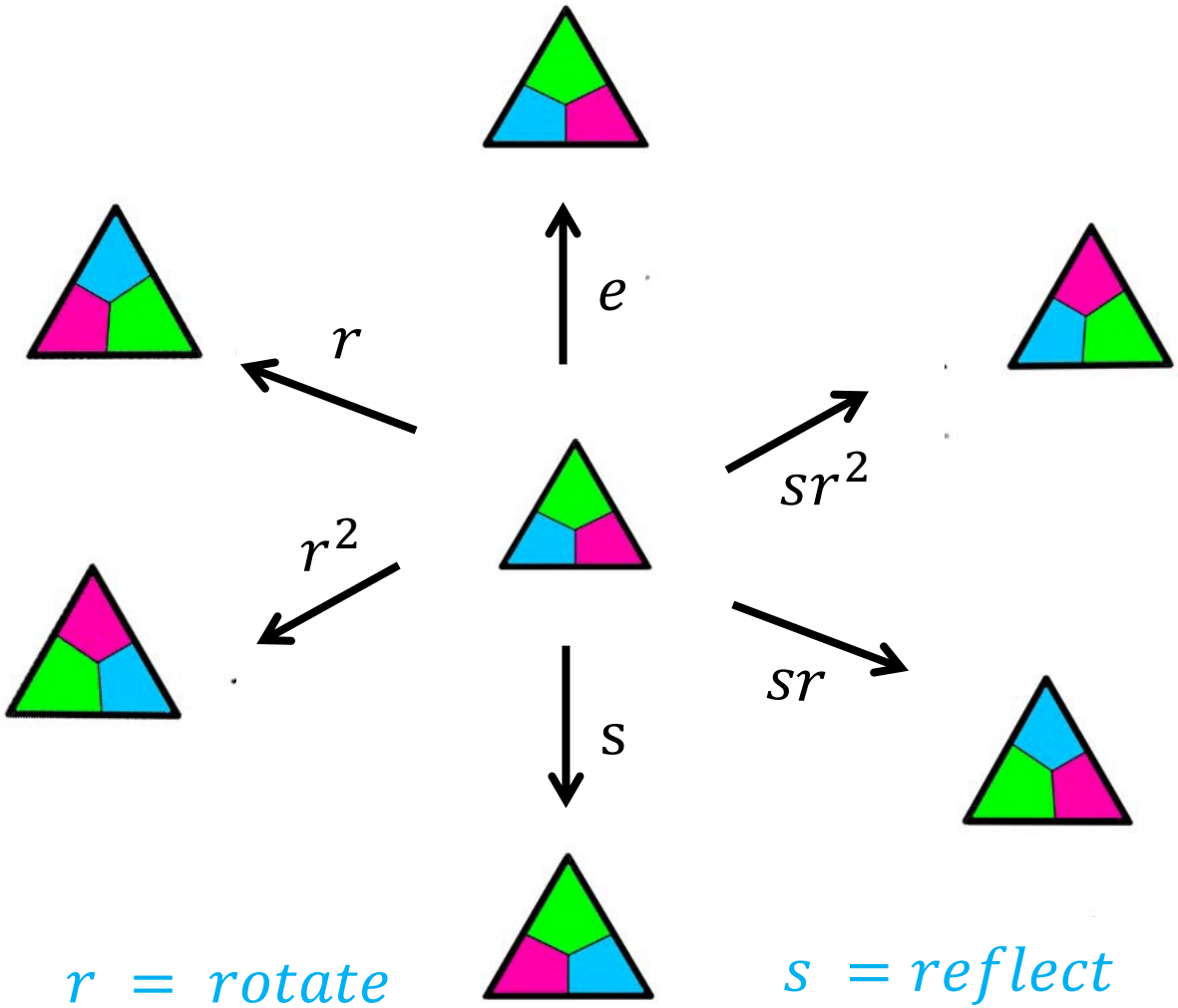
D3の r, sに次のようなRとSを対応させる。（Functorだ！）

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

Rは、右回りに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。

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

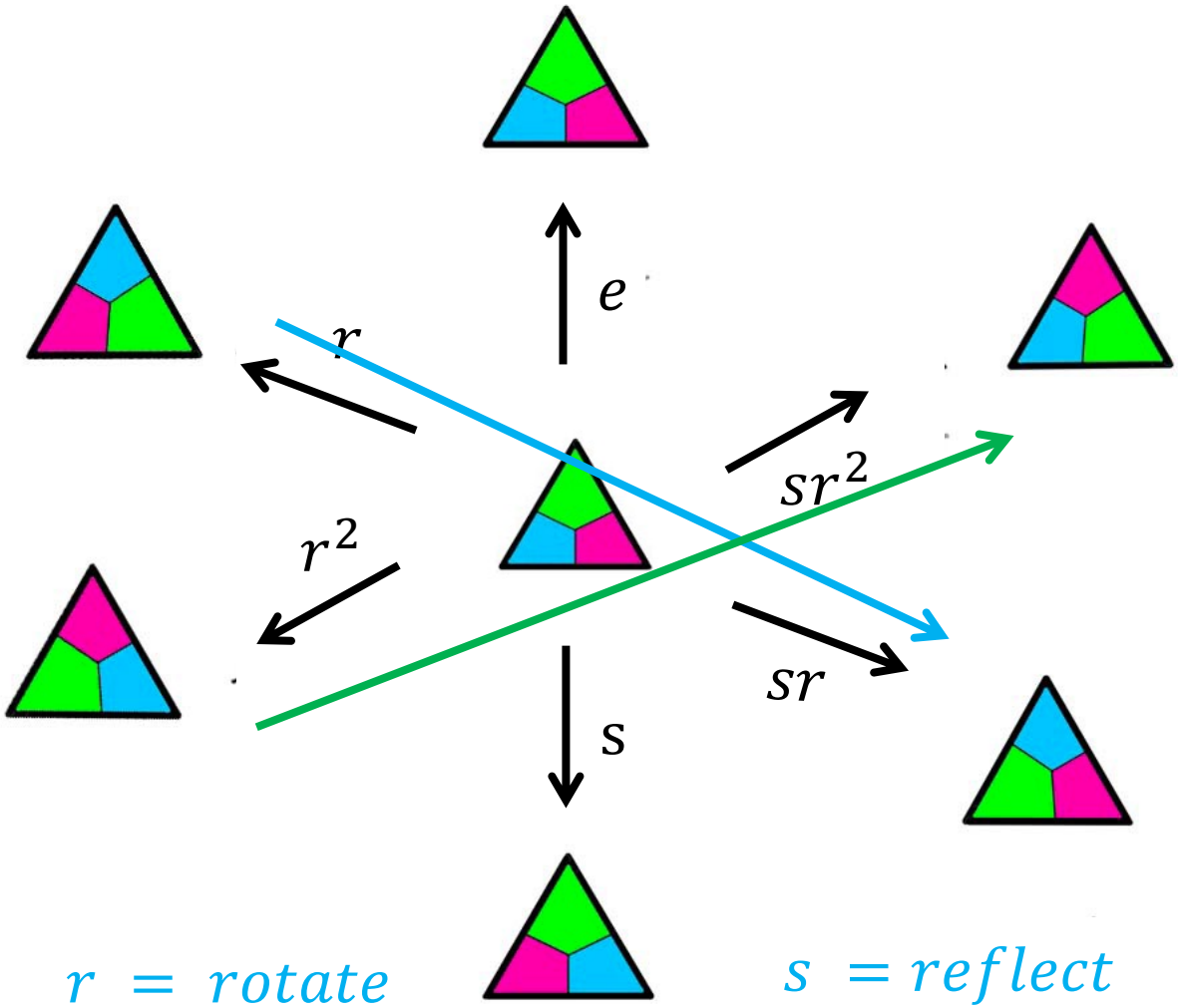


r = rotate

s = reflect

"What is applied category theory?"
https://arxiv.org/pdf/1809.05923.pdf

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

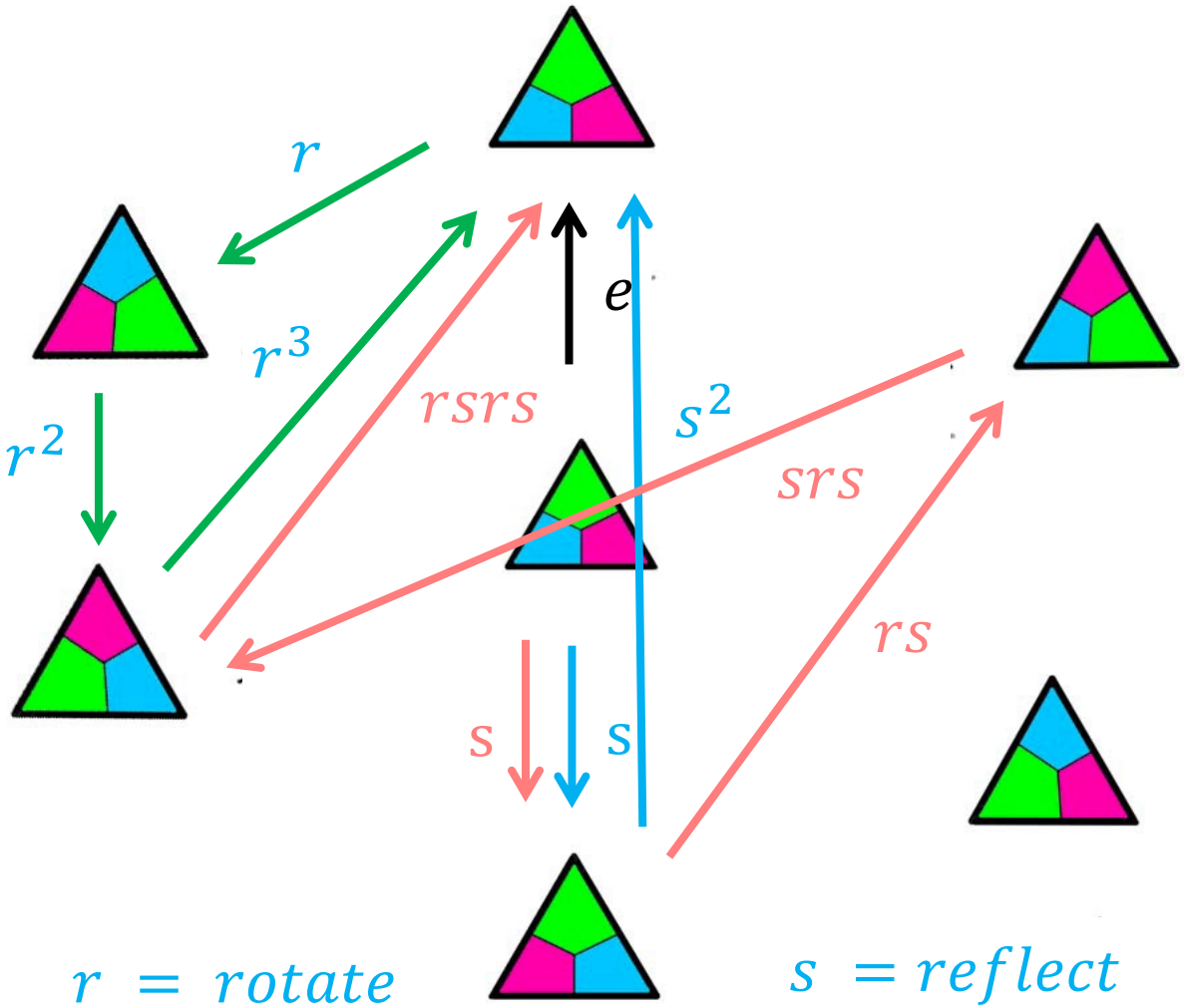


r = rotate

s = reflect

"What is applied category theory?"
https://arxiv.org/pdf/1809.05923.pdf

$$D_3 = < r, s \mid r^3 = s^2 = rsrs = 1 >$$



"What is applied category theory?"
https://arxiv.org/pdf/1809.05923.pdf

Functorial Semantics

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, …を集合として、射 f, g, h, … 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、

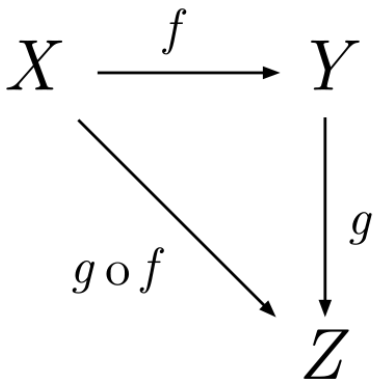
$$f : A \rightarrow B$$

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 → Y, g: Y → Z に対して、射の「合成」
   g∘f: X → Zが存在する。



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

Functor

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f射を Dの射F(f) に

対応づけるFをFunctorと呼ぶ。

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

Cの射 $f\colon X \to Y$ を Dの射 $F(f)\colon F(Y) \to F(X)$ 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

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

1963年

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。

$$Functor$$

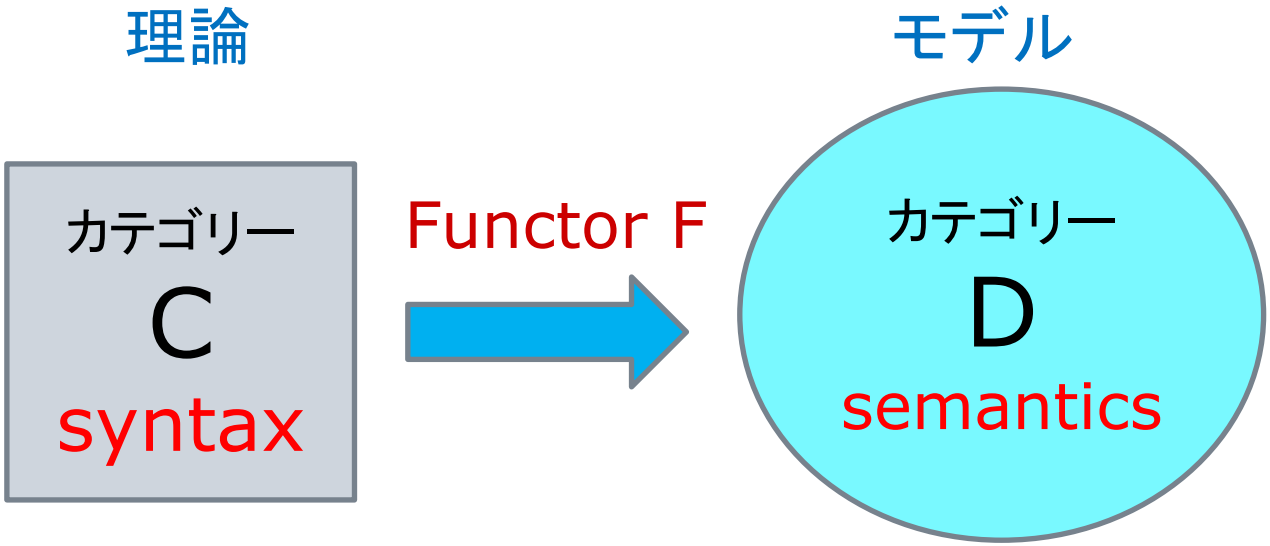
$$F$$

$$C \rightarrow D$$

$$Syntax \rightarrow Semantics$$

LawvereのFunctorial Semantics

Functor Fは、Syntax → 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。



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。そ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!

Part 2

ことばの構成性
Pregroup Grammar

Part 2

ことばの構成性
Pregroup Grammar

- ChomskyとLambek
- Combinatory Categorial Grammar
- Pregroup Grammar

ことばと意味の構成性

ことばは、より基本的な要素からあるルールに従って構成されています。これをことばの「構成性」と言います。少し単純化して、この基本的な要素を「語」、そのルールを「文法」と呼ぶことにしましょう。

こうした観察は、20世紀の「生成文法」が初めてのものではありません。驚くべきことに、紀元前4世紀に、インドのパーニニは、精緻なサンスクリット文法の記述を成し遂げていたと言います。

意味もまた、より基本的なものから、何らかのルールに従って構成されていると考えることができます。

ここでは、ことばと意味の「構成性」についての、基本的な観察をまとめてみましょう。

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.

ことばの構成性 – 文法

このセッションでは、ことばの構成性を示す文法理論の現代版として、LambekのPregroup Grammar を紹介します。

ChomskyとLambek

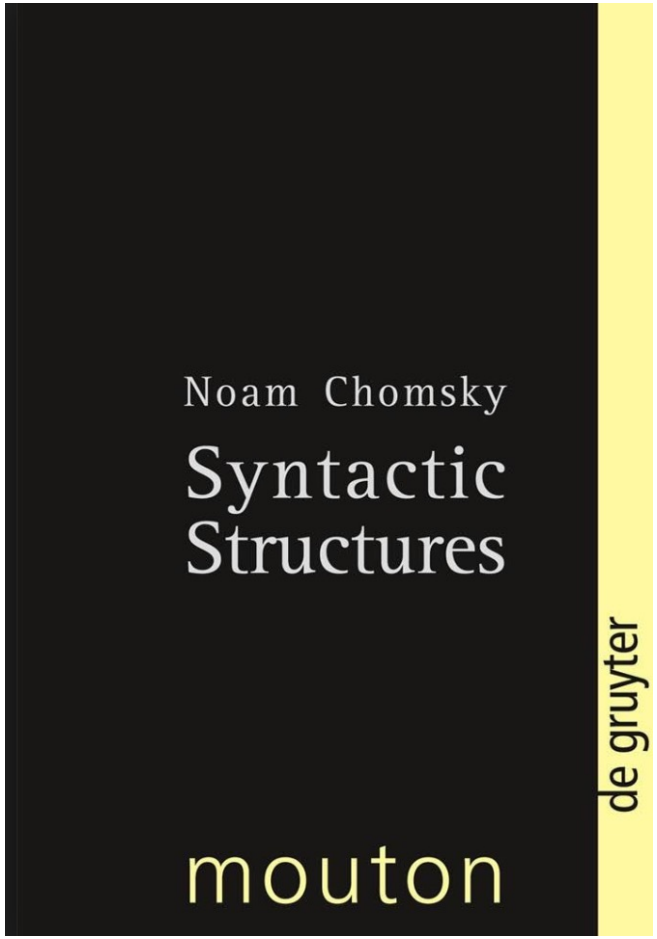
Noam Chomsky

1928 –

Joachim Lambek

1922 – 2014

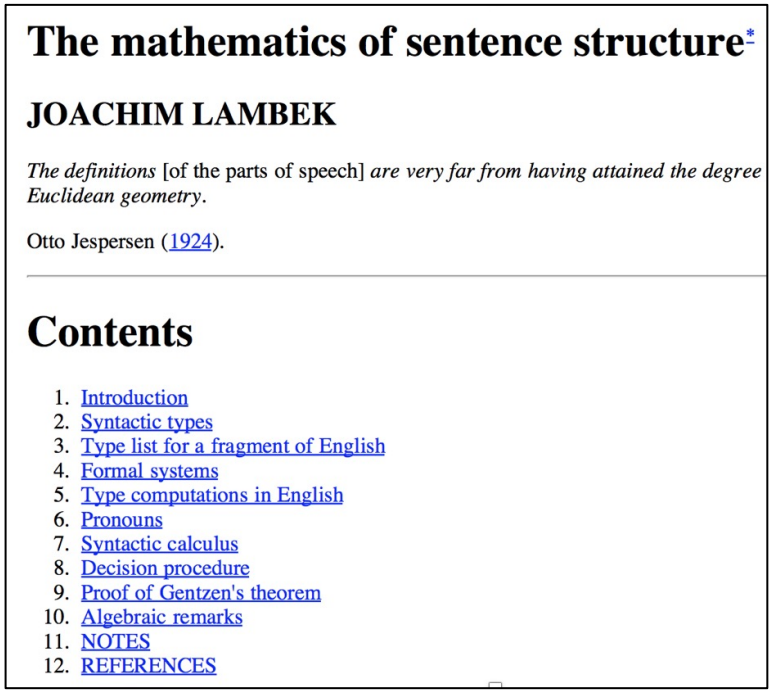
Chomsky 1957年
”Syntactic Structures”



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

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 --> x

y(y\x) --> 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してみせる。

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

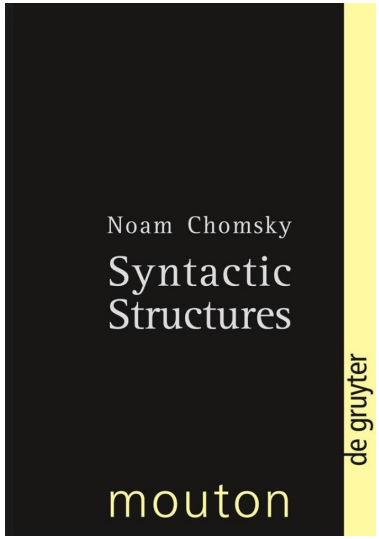
Chomsky, Lambek 再び出会う

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Categirial 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” を発行するする。

40年以上の時を隔てて、両者は再会する。

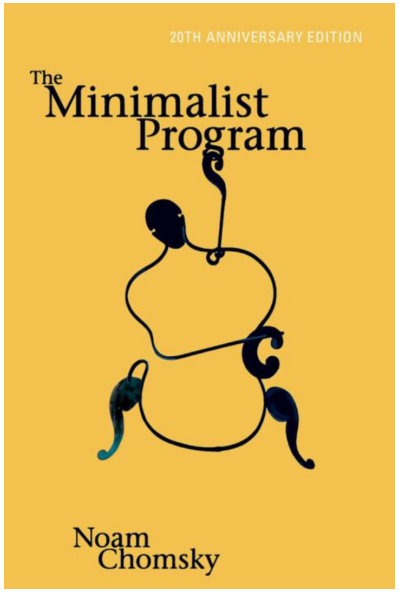
1957年



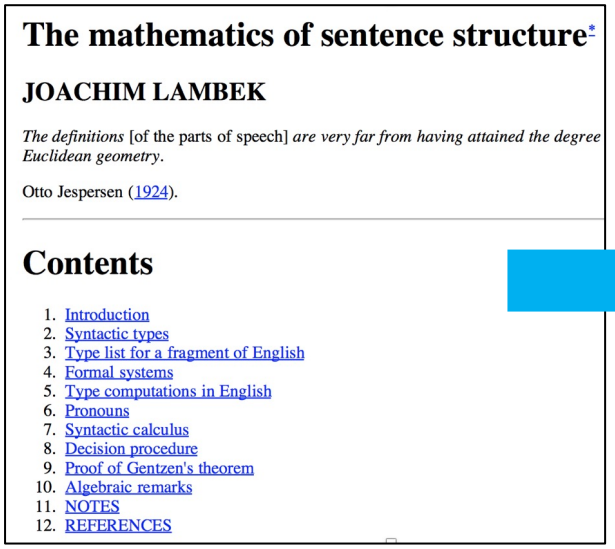
Chomsky

48年

1995年



1958年



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).

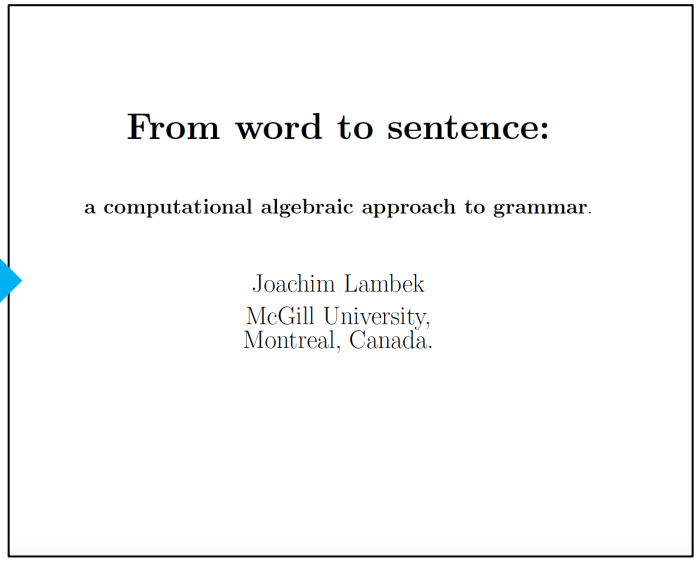
Contents

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

Lambek

50年！

2008年



From word to sentence:

a computational algebraic approach to grammar.

Joachim Lambek
McGill University,
Montreal, Canada.

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。」

Lambek:“Word to Sentence” 36. Postscript on Minimalism.

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

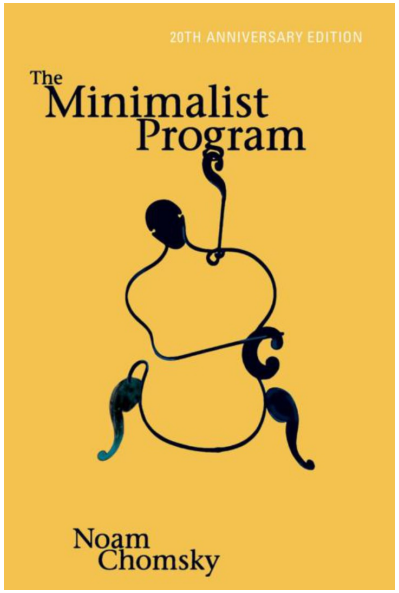
Imagine the following scene. You are at your favorite beer hall somewhere in Amsterdam—let's call it the Cartesian Well. Well known meeting place of intelligentsia, you are not surprised when a thin person dressed all in black sidles up to you and whispers in your ear, "Have I got a linguistic theory 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 Categirial 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

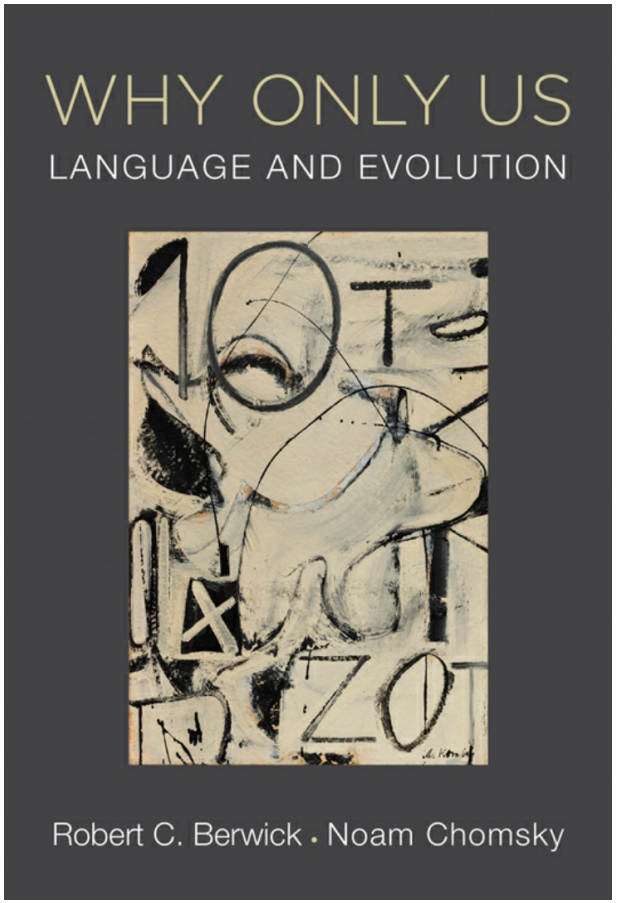
ABSTRACT

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 (alpha, beta), where alpha is a subtree of beta; this automatically derives the usual c-command condition on so-called "empty categories." The usual semantic interpretation benefits of the categorial terpret" 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 Categorial Grammar

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、語の連接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

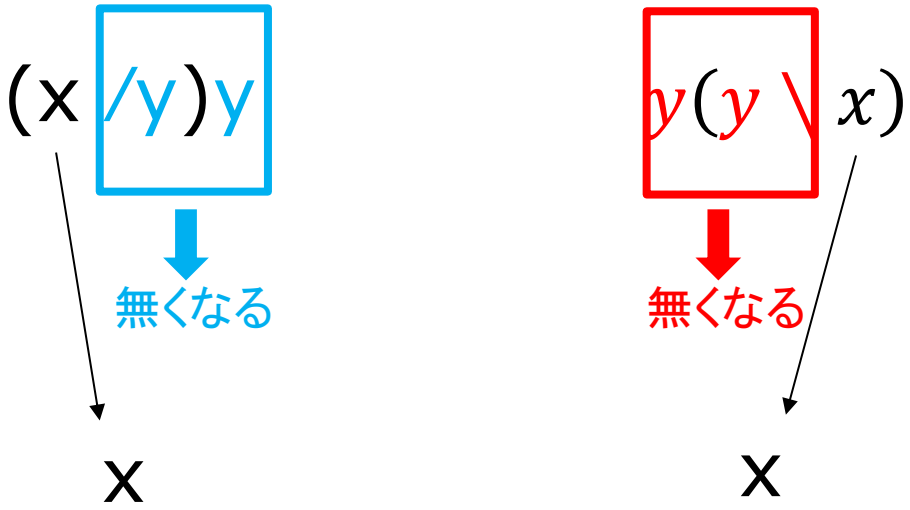
Syntactic types

- 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, x under y と読む。

Syntactic types

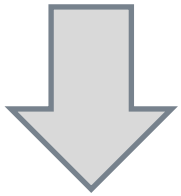
$$(\mathrm{I}) \quad (x/y)y \rightarrow x, \quad y(y\backslash x) \rightarrow x.$$



Syntactic types

$$\text{(I)} \quad (x/y)y \to x, \quad y(y\backslash x) \to x.$$

1958年の定式化



2008年の定式化

$$xx^r \to 1, \quad x^\ell x \to 1,$$

Sample 1

John works

John Works

$n \quad n \setminus s$

n(n\s)

n(n\s)

s

Sample 2
Poor John works

$(Poor\ John)\ works$

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

(n/n)n(n\s)

(n/n)n(n\s)

n(n\s)

s

Sample 3
John works here

$(John\ \ works)\ here$

$n \qquad n \setminus s \qquad s \setminus s$

n(n\s) (s\s)

n(n\s) (s\s)

s(s\s)

s

Sample 4

John never works

John (*never* *works*)

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

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

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

n(n\s)

s

Sample 5
John works for Jane

$(John\ \ works)$ $(for\ \ Jane)$

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

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

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

s(s\s)

s

Sample 6

John works and Jane rests

$(John\ works)\ (and\quad (Jane\ rests))$

$(\ n\quad (\mathrm{n}\backslash\mathrm{s}))\quad ((\mathrm{s}\backslash\mathrm{s})\ /\ s\ \ (\ n\ \ (\mathrm{n}\backslash\mathrm{s})))$

$(n(\mathrm{n}\backslash\mathrm{s}))((\mathrm{s}\backslash\mathrm{s})\ /\ s(\ n(\mathrm{n}\backslash\mathrm{s})))$

$(n(\mathrm{n}\backslash\mathrm{s}))((\mathrm{s}\backslash\mathrm{s})\ /\ s(\ n(\mathrm{n}\backslash\mathrm{s})))$

$\mathrm{s}(\mathrm{s}\backslash\mathrm{s})\ /\ s(\mathrm{s}))$

s

Sample 7

John likes Jane

John (*likes* *Jane*)

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

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

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

n (n\s)

s

語の型と品詞

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

Pregroup Grammar

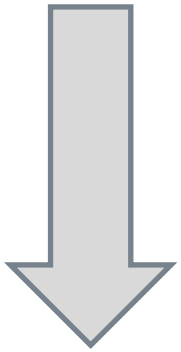
From word to sentence:
a computational algebraic approach to grammar

Joachim Lambek

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

Syntactic types

$$\text{(I)} \quad (x/y)y \rightarrow x, \quad y(y\backslash x) \rightarrow x.$$

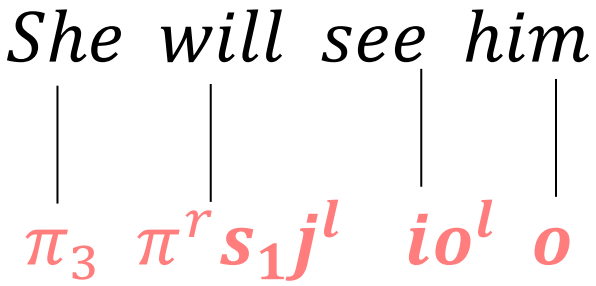


1958年の定式化

2008年の定式化

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

“She will see him” の解析



$$She : \pi_3$$
$$will : \pi^r s_1 j^l$$
$$see : io^l$$
$$him : o$$

先の例に出ている「基本的な型」

$$She\ \ will\ \ see\ \ him$$

$$\pi_3\left(\pi^r \boldsymbol{s_1 j^l}\right)\left(\boldsymbol{i o^l}\right)\boldsymbol{o}$$

π ＝ 主語

π_3 ＝ 三人称単数の主語

$\boldsymbol{s_1}$ ＝ 現在形の宣言文

$\boldsymbol{i}$ ＝ 自動詞の不定形

$\boldsymbol{j}$ ＝ 完全動詞句の不定形

$\boldsymbol{o}$ ＝ 直接目的語

半順序 $\rightarrow$

$\rightarrow$ 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。

$\pi_3 \rightarrow \pi, \boldsymbol{i} \rightarrow \boldsymbol{j}$

x→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 $\pi_3 \rightarrow \pi,\ \boldsymbol{i} \rightarrow \boldsymbol{j}$ のもとで、

$$She\ will\ see\ him$$
$$\pi_3(\pi^r \boldsymbol{s_1} j^l)(io^l)o$$
$$= [\pi_3 \pi^r]\boldsymbol{s_1}[j^l i][o^l o]$$
$$\rightarrow [\pi\pi^r]\boldsymbol{s_1}[j^l \boldsymbol{j}][o^l o]$$
$$\rightarrow 1\ s_1\ 1\ 1 \rightarrow\ s_1$$

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

$$\pi_3(\pi^r \mathbf{s}_1 \mathbf{j}^\ell)(\mathbf{i}\mathbf{o}^\ell)\mathbf{o} = [\pi_3\pi^r]\mathbf{s}_1\,[\mathbf{j}^\ell\mathbf{i}][\mathbf{o}^\ell\mathbf{o}]$$
$$\to [\pi\pi^r]\mathbf{s}_1\,[\mathbf{j}^\ell\mathbf{j}]\,[\mathbf{o}^\ell\mathbf{o}]$$
$$\to 1\mathbf{s}_1\,1\,1 \;=\; \mathbf{s}_1,$$

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

$$xx^r \to 1, \quad x^\ell x \to 1,$$

$$1x = x = x1,$$

"she will come"

She will come

$\pi_3 \ \pi^r \boldsymbol{s}_1 \boldsymbol{j}^l \ \ \boldsymbol{i}$

will: $\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l$

$\pi_3 (\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l) \boldsymbol{i}$

$\rightarrow \ \boldsymbol{\pi} \ (\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l) \boldsymbol{i}$

$\rightarrow \ \pi \ (\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l) \ \boldsymbol{j}$

$\rightarrow \ \boldsymbol{\pi} \ \boldsymbol{\pi}^r \ \boldsymbol{s}_1 \ \boldsymbol{j}^l \ \boldsymbol{j}$

$\rightarrow 1 \boldsymbol{s}_1 1$

$\rightarrow \ \boldsymbol{s}_1$

"She will see him"

She will see him

$\pi_3 \;\; \pi^r \boldsymbol{s}_1 \boldsymbol{j}^l \;\; \boldsymbol{i}\boldsymbol{o}^l \;\; \boldsymbol{o}$

$\pi_3 \, (\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l)(\boldsymbol{i}\boldsymbol{o}^l)\boldsymbol{o}$

$\rightarrow \; \boldsymbol{\pi}\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l \boldsymbol{i}\boldsymbol{o}^l \boldsymbol{o}$

$\rightarrow \; \pi\pi^r \boldsymbol{s}_1 \boldsymbol{j}^l \, \boldsymbol{j}\boldsymbol{o}^l \boldsymbol{o}$

$\rightarrow \; (\pi\pi^r)\boldsymbol{s}_1(\boldsymbol{j}^l \, \boldsymbol{j})(\boldsymbol{o}^l \boldsymbol{o})$

$\rightarrow \; 1\boldsymbol{s}_1 11$

$\rightarrow \; \boldsymbol{s}_1$

“Will she see him?”

$$Will\ she\ see\ him\ ?$$

$$(q_1 j^l \pi_l)\pi_3(io^l)o \rightarrow \boldsymbol{q_1}$$

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

ここで、新しい型 $\boldsymbol{q_1}$を導入する

$\boldsymbol{q_1}$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?"

$$Whom\ will\ she\ see\ ?$$

$$(\bar{q}o^{ll}q^{l})(q_1 j^{l}\pi^{l})\pi_3(io^{l}) \rightarrow \bar{q}$$

ここで、新しい型 q と$\bar{q}$を導入する

$\boldsymbol{q}$は、yesかno で答える任意の時制の疑問文

$\bar{\boldsymbol{q}}$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、$x^{ll}x^{l} \rightarrow 1,$ $\quad x^{r}x^{rr} \rightarrow 1$

(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)

$$\begin{array}{l} \textit{whom will she see} - ? \\ (\overline{\mathbf{q}}\mathbf{o}^{\ell\ell}\mathbf{q}^{\ell})(\mathbf{q}_1\mathbf{j}^{\ell}\pi^{\ell})\pi_3(\mathbf{i}\mathbf{o}^{\ell}) \quad \rightarrow \quad \overline{\mathbf{q}} \end{array}$$

$$\begin{array}{rl} \textit{whom}: & \overline{\mathbf{q}}\hat{\mathbf{o}}^{\ell\ell}\mathbf{q}^{\ell}, \\ \textit{whom will}: & \overline{\mathbf{q}}\hat{\mathbf{o}}^{\ell\ell}\,\underline{\mathbf{q}^{\ell}\mathbf{q}_1}\mathbf{j}^{\ell}\pi^{\ell} \;\rightarrow\; \overline{\mathbf{q}}\hat{\mathbf{o}}^{\ell\ell}\mathbf{j}^{\ell}\pi^{\ell}, \\ \textit{whom will she}: & \overline{\mathbf{q}}\hat{\mathbf{o}}^{\ell\ell}\mathbf{j}^{\ell}\pi^{\ell}\pi_3 \;\rightarrow\; \overline{\mathbf{q}}\hat{\mathbf{o}}^{\ell\ell}\mathbf{j}^{\ell}, \\ \textit{whom will she see}: & \overline{\mathbf{q}}\hat{\mathbf{o}}^{\ell\ell}\,\underline{\mathbf{j}^{\ell}\,\overbrace{\mathbf{i}\,\mathbf{o}^{\ell}}} \;\rightarrow\; \overline{\mathbf{q}}\,\underline{\hat{\mathbf{o}}^{\ell\ell}\mathbf{o}^{\ell}} \;\rightarrow\; \overline{\mathbf{q}}. \end{array}$$

Part 3

カテゴリー論的構成的分散意味論

DisCoCatの登場

ここでは、CoeckeのDisCoCat論文を紹介する。

Part 3-1

カテゴリー論的構成的分散意味論

DisCoCatの登場

Bob Coecke

- Monoidal categoryとString Diagram
- 語の意味から文の意味へ」のプロセスを図形で表す
- 「語の意味から文の意味へ」の写像

カテゴリー論的
構成的分散意味論

ここでは、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。

Mathematical Foundations for a Compositional Distributional Model of Meaning

Bob Coecke, Mehrnoosh Sadrzadeh,
Stephen Clark

https://arxiv.org/abs/1003.4394

2010年

Around 2008 @ Oxford Uni. there were 3 people:

- One knew grammar algebra:

Mehrnoosh Sadrzadeh

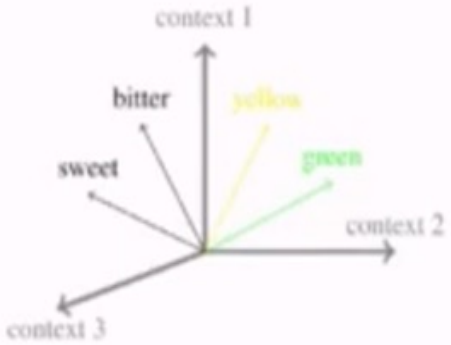
LambekのPregroupだ

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

- One knew vector-space NLP:

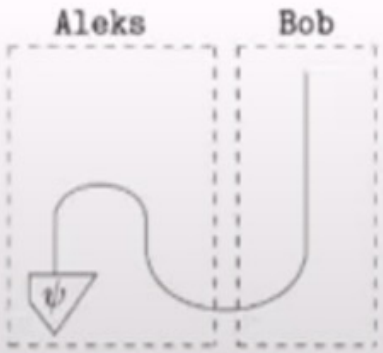
Stephen Clark

“Word2Vec”



- One (=me) knew categorical QM:

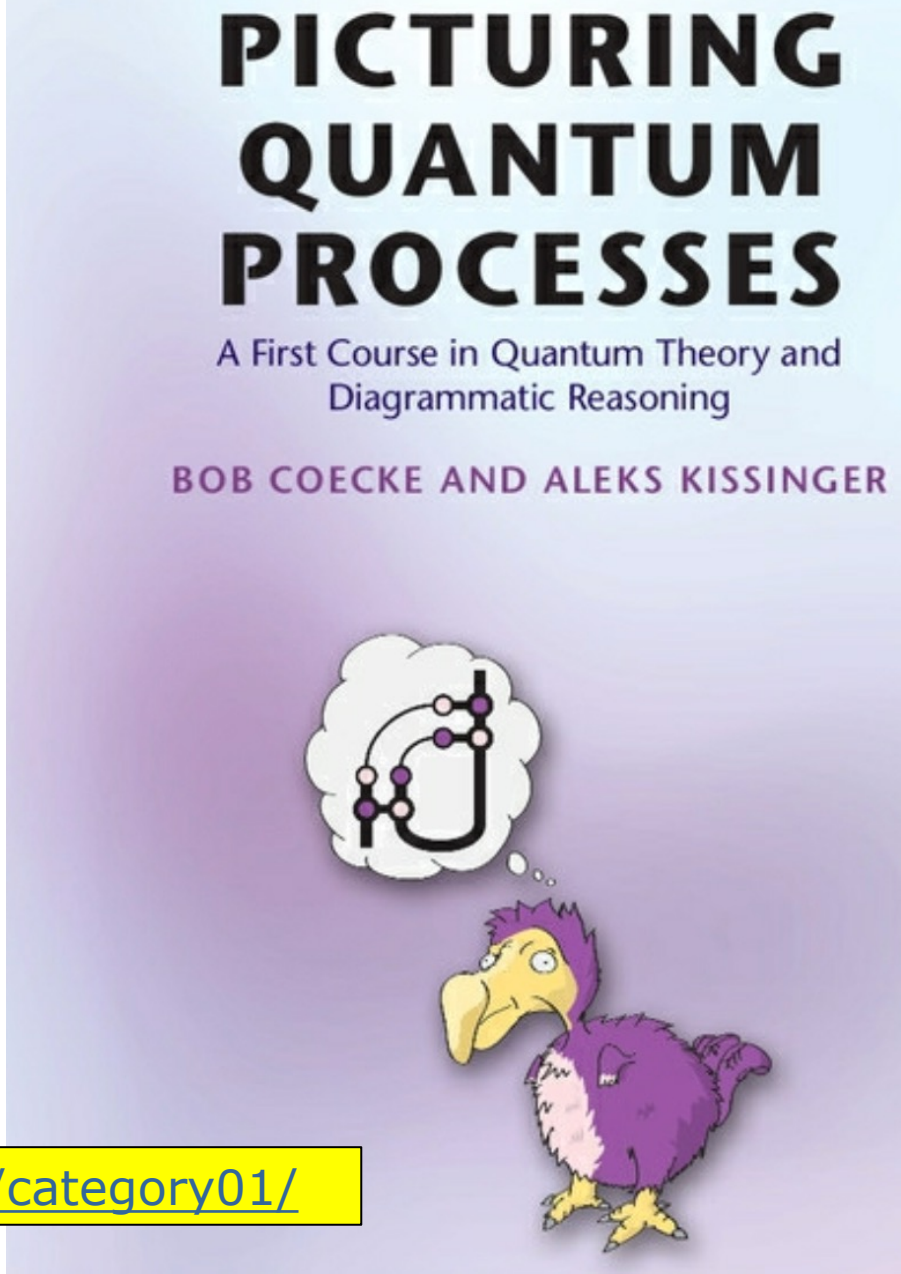
Bob Coecke



Bob Coecke

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/

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。

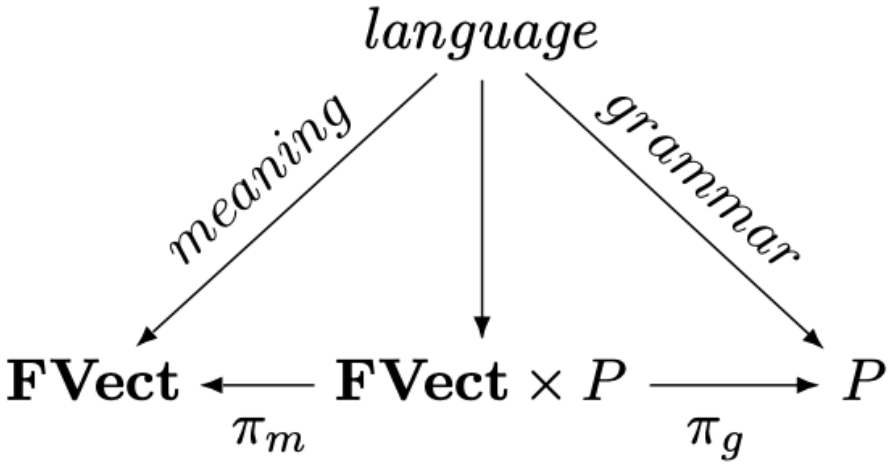
我々が用いる数学的構造は、文全体の意味を作り出すために、文の中の単語間の情報の流れを明らかにする、純粋にダイアグラム的な計算を可能にする。

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カテゴリーの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カテゴリーとなる。

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。

Monoidal categoryとString Diagram

意味の構成的モデルを図形で表す

この論文の大きな特徴の一つは、意味の構成的モデルを、図形で表現していることです。

そこでは、Monoidal CategoryのString Diagram による表現が用いられています。

その特徴を簡単に見ておきましょう。

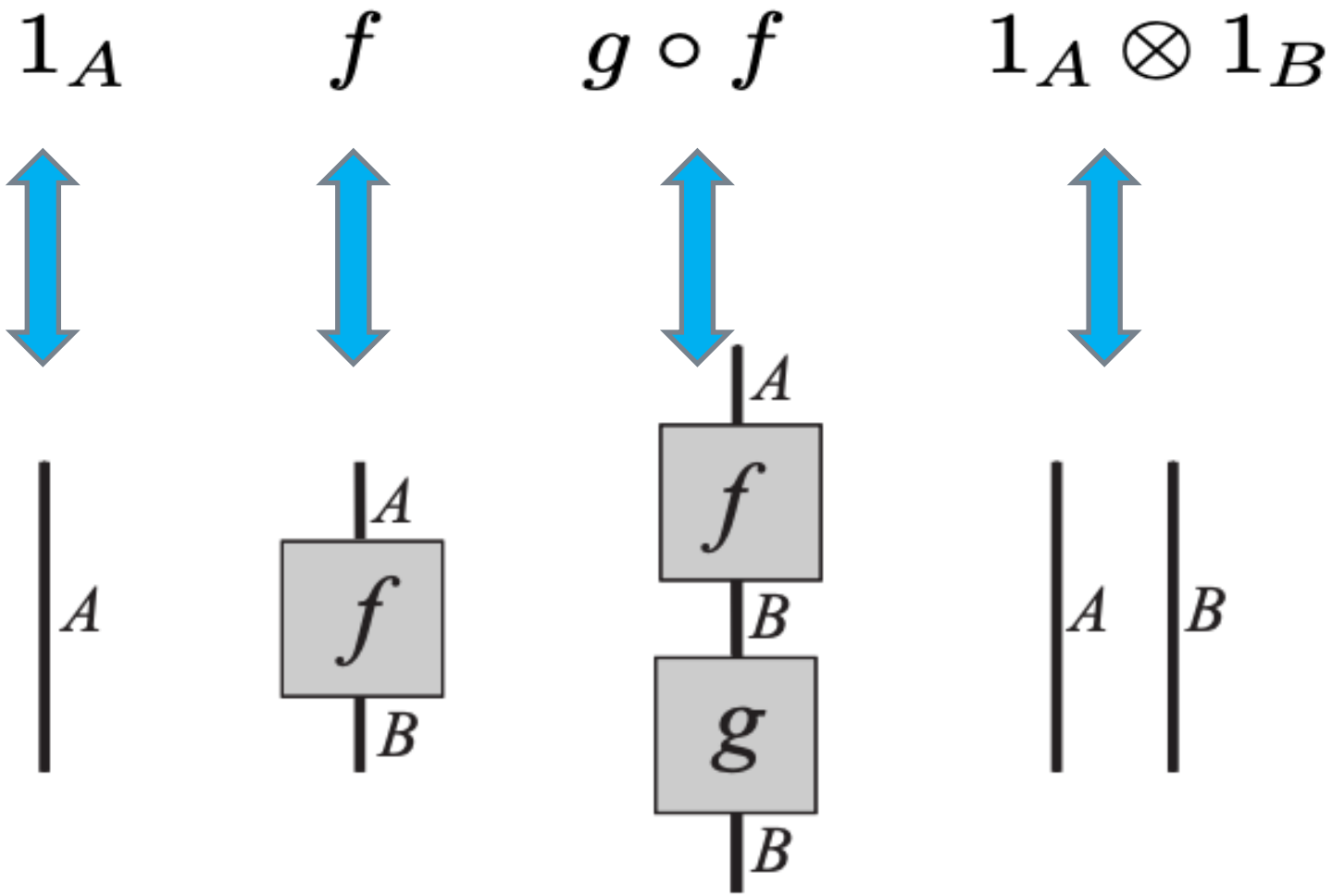
数学的証明＝図形的導出

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。

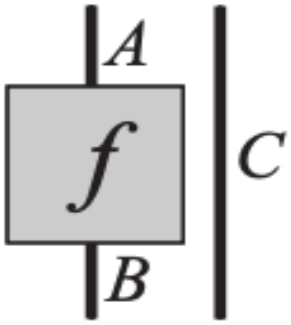
すなわち、String Diagramで図形的に導出可能な等式は、monoidal カテゴリーの公理から証明可能である。

式と図形との基本的な対応

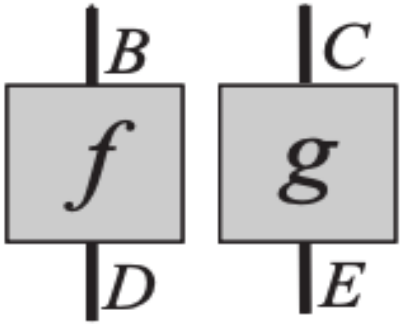


式と図形との基本的な対応

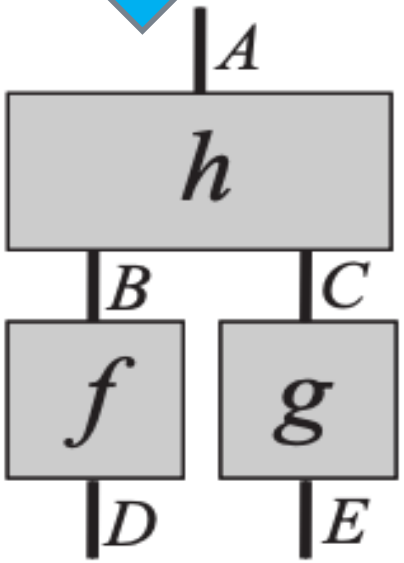
$f \otimes 1_C$



$f \otimes g$



$(f \otimes g) \circ h$

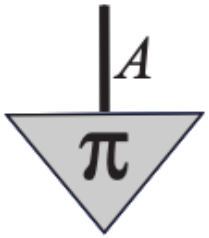


式と図形との基本的な対応

$\psi : \mathrm{I} \to A$

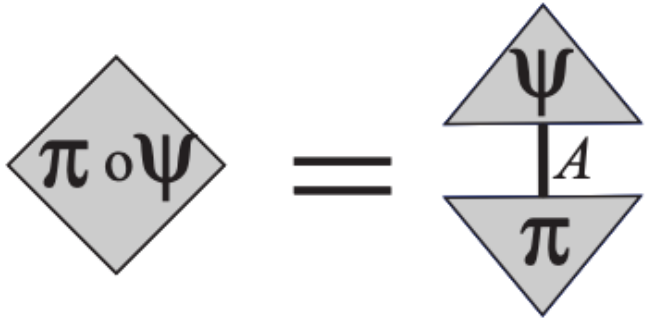
入力を持たず
出力のみを持つ
「状態」

$\pi : A \to \mathrm{I}$



出力を持たず
入力のみを持つ
「効果・テスト」

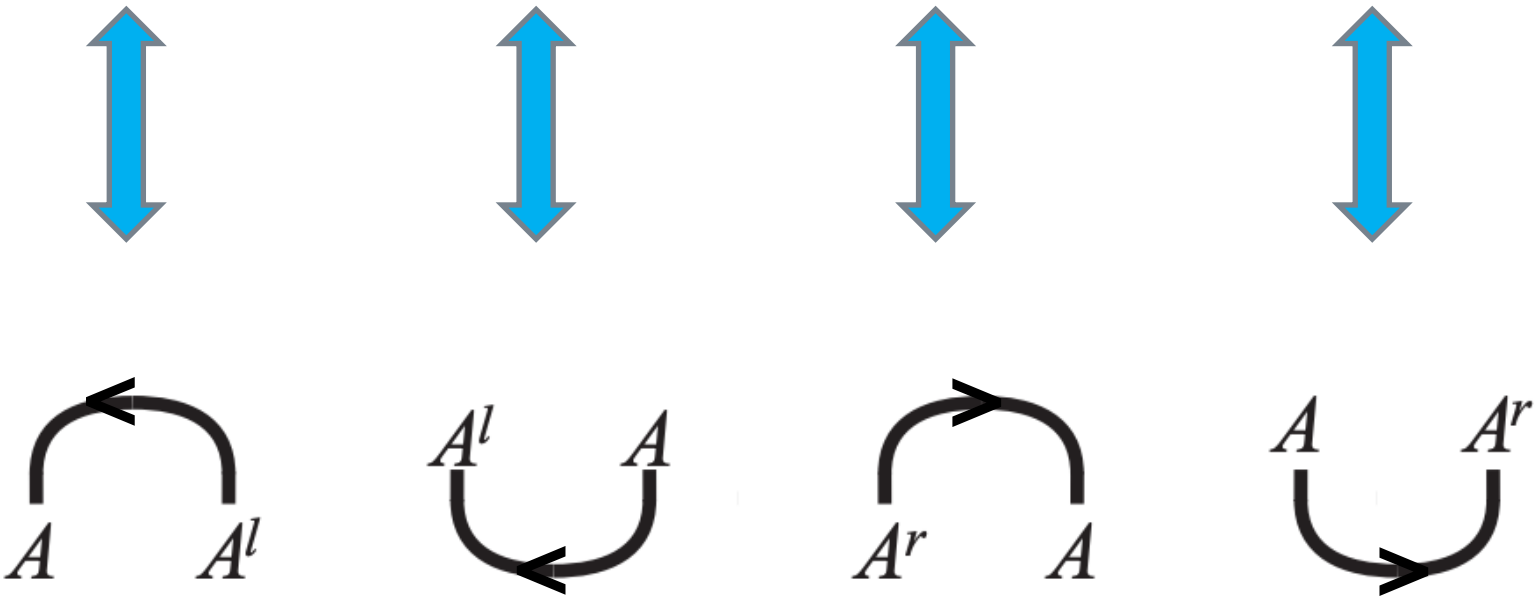
$\pi \circ \psi : \mathrm{I} \to \mathrm{I}$



“Picturing Quantum Processes”と逆のnotationになっている
ここでは情報は上から下に流れると想定されているから・

式と図形との基本的な対応

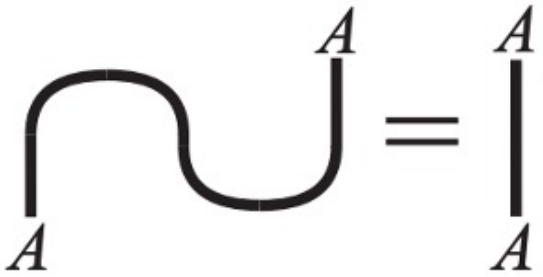
$\eta^l : \mathrm{I} \to A \otimes A^l$　$\epsilon^l : A^l \otimes A \to \mathrm{I}$　$\eta^r : \mathrm{I} \to A^r \otimes A$　$\epsilon^r : A \otimes A^r \to \mathrm{I}$

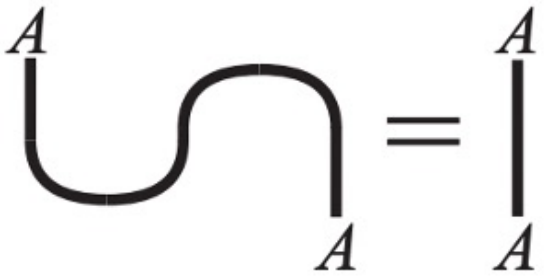


η^l　ϵ^l　η^r　ϵ^r

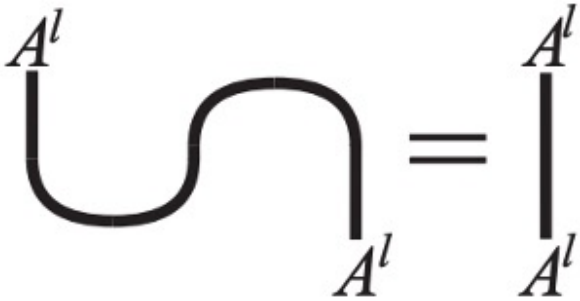
図形を変形する等式
Yanking 等式

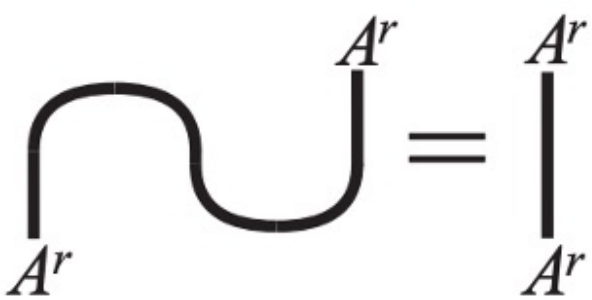
$$(1_A \otimes \epsilon^l) \circ (\eta^l \otimes 1_A) = 1_A \qquad (\epsilon^r \otimes 1_A) \circ (1_A \otimes \eta^r) = 1_A$$



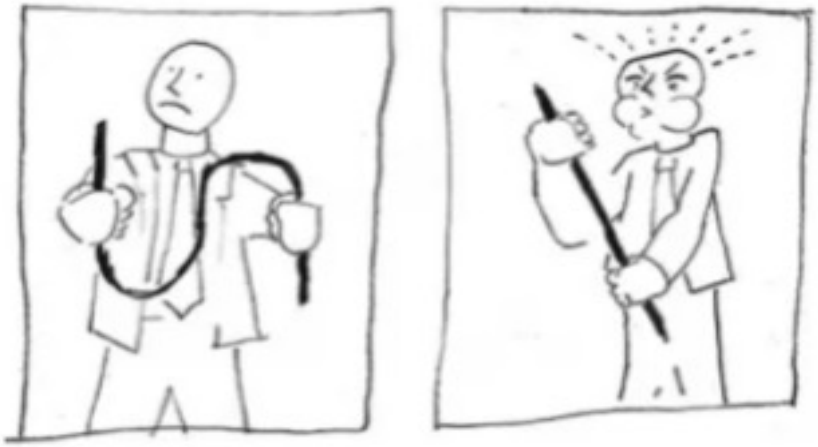


$$(\epsilon^l \otimes 1_{A^l}) \circ (1_{A^l} \otimes \eta^l) = 1_{A^l} \qquad (1_{A^r} \otimes \epsilon^r) \circ (\eta^r \otimes 1_{A^r}) = 1_{A^r}$$





String Diagram については
次のページを参照してほしい

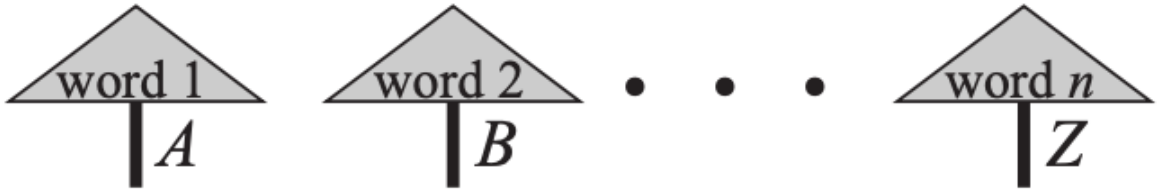


String Diagram を学ぶ
-- カテゴリー論入門 (1)

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

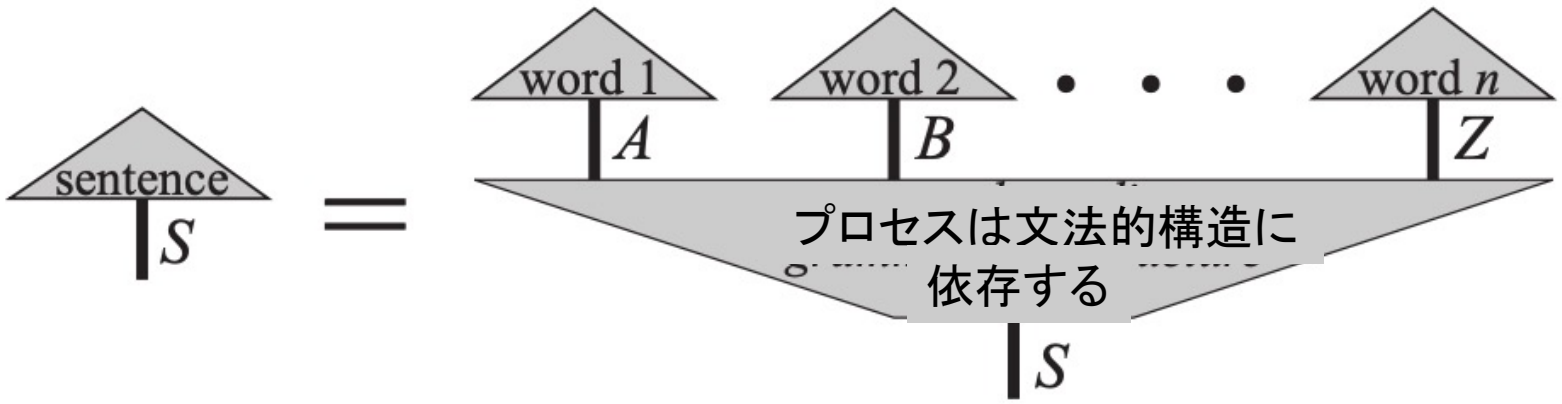
「語の意味から文の意味へ」の
プロセスを図形で表す

「語の意味」の並び

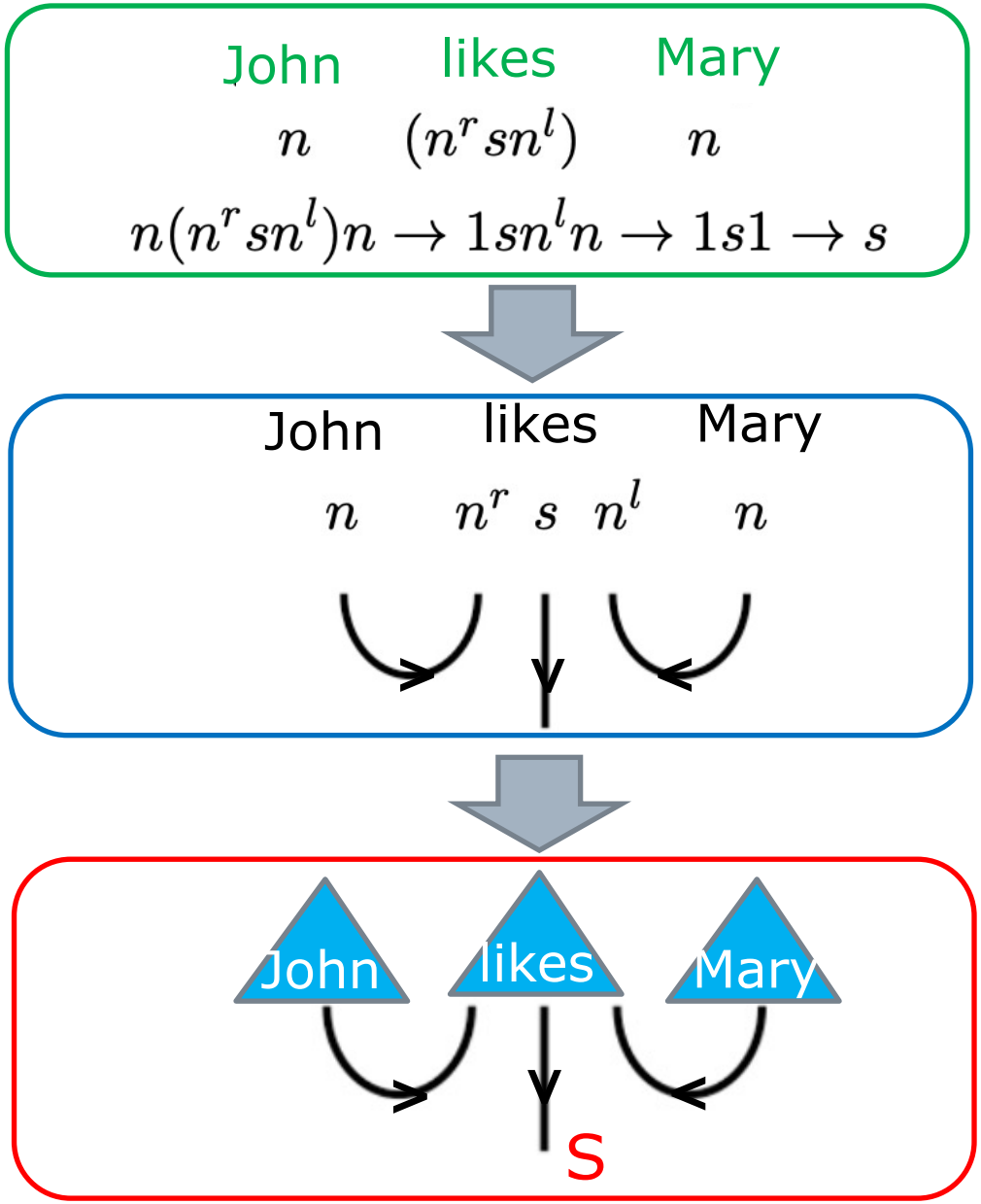


「文の意味」

「語の意味から文の意味へ」のプロセス



“John likes Mary” の意味 S の図形表現



Pregroupの型

Pregroupでのsの導出

文 s の導出をString Diagramで表現

語の意味表現の並び

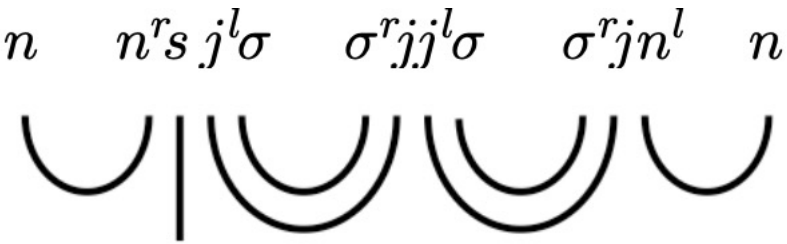
文の意味表現 S

“John does not like Mary” の意味を図形で表現する

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
j: infinitive of the verb
s: declarative statement
σ: glueing type

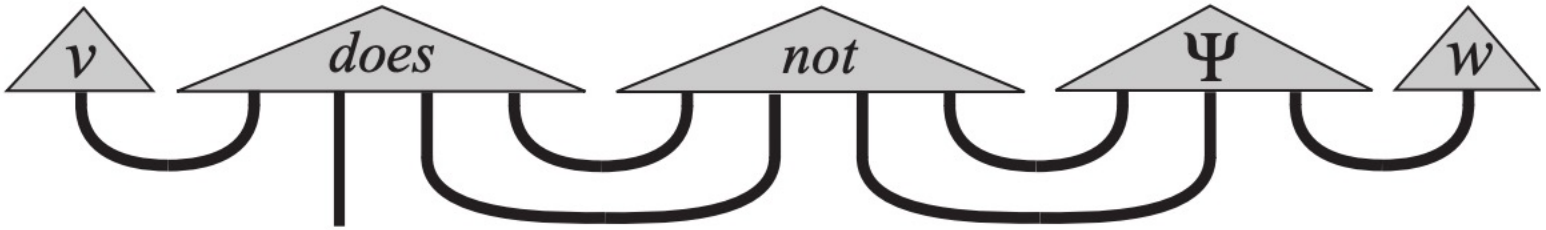
$$n\,(n^r s j^l \sigma)\,(\sigma^r j j^l \sigma)\,(\sigma^r j n^l)\,n \to s$$



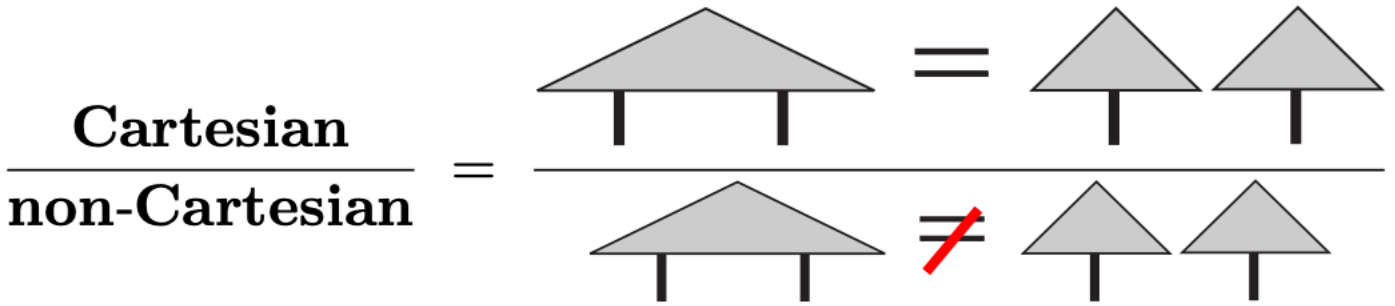
“John does not like Mary” の意味を図形で表現する

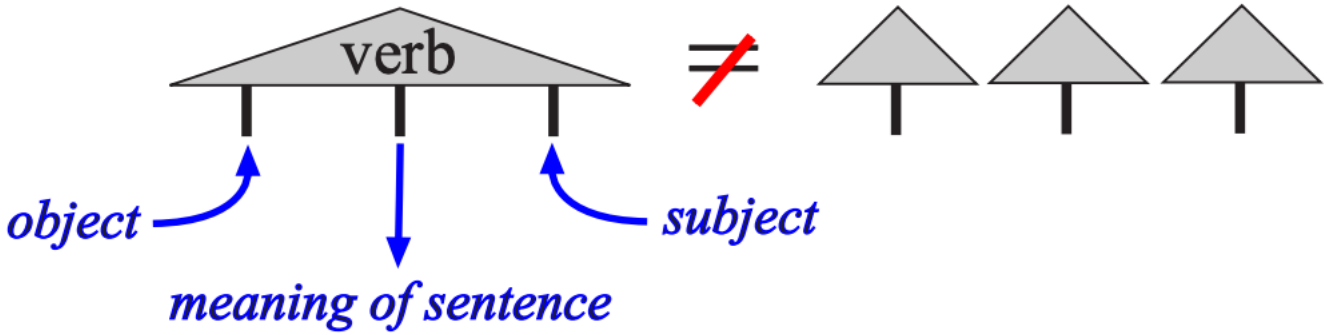
$$f = (1_S \otimes \epsilon_J \otimes \epsilon_J) \circ (\epsilon_V \otimes 1_S \otimes 1_{J^*} \otimes \epsilon_V \otimes 1_J \otimes 1_{J^*} \otimes \epsilon_V \otimes 1_J \otimes \epsilon_W) :$$
$$V \otimes (V^* \otimes S \otimes J^* \otimes V) \otimes (V^* \otimes J \otimes J^* \otimes V) \otimes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

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



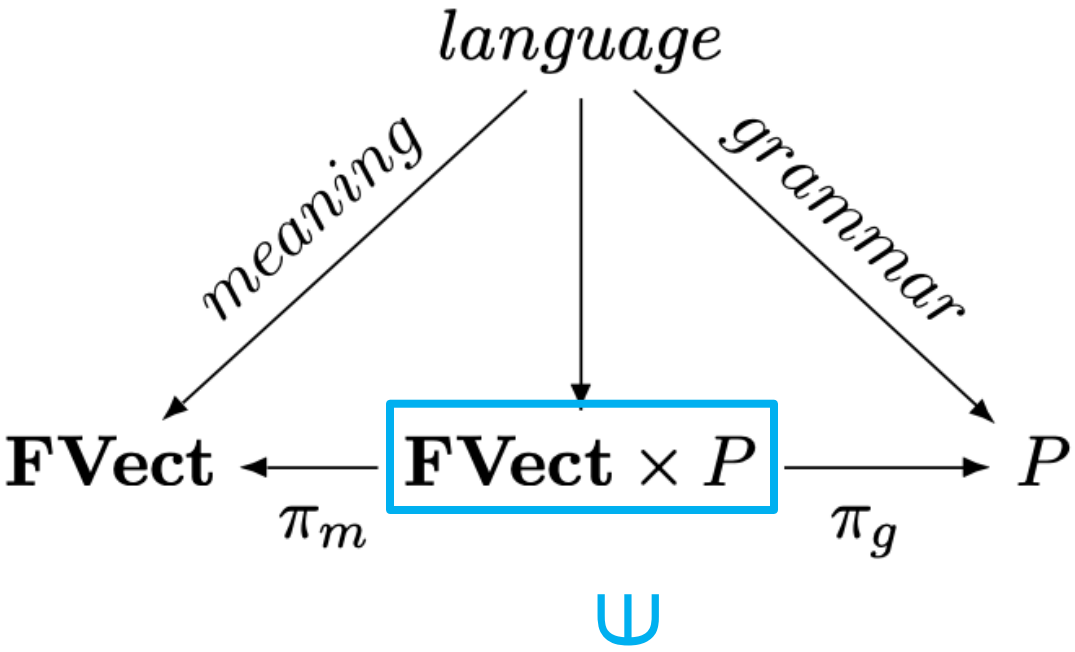
「分離不能」なプロセスとしての「文の意味」





「語の意味から文の意味へ」の写像

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



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、$\alpha$中の$p_i$を$W_i$に置き換えることで作ることができる。

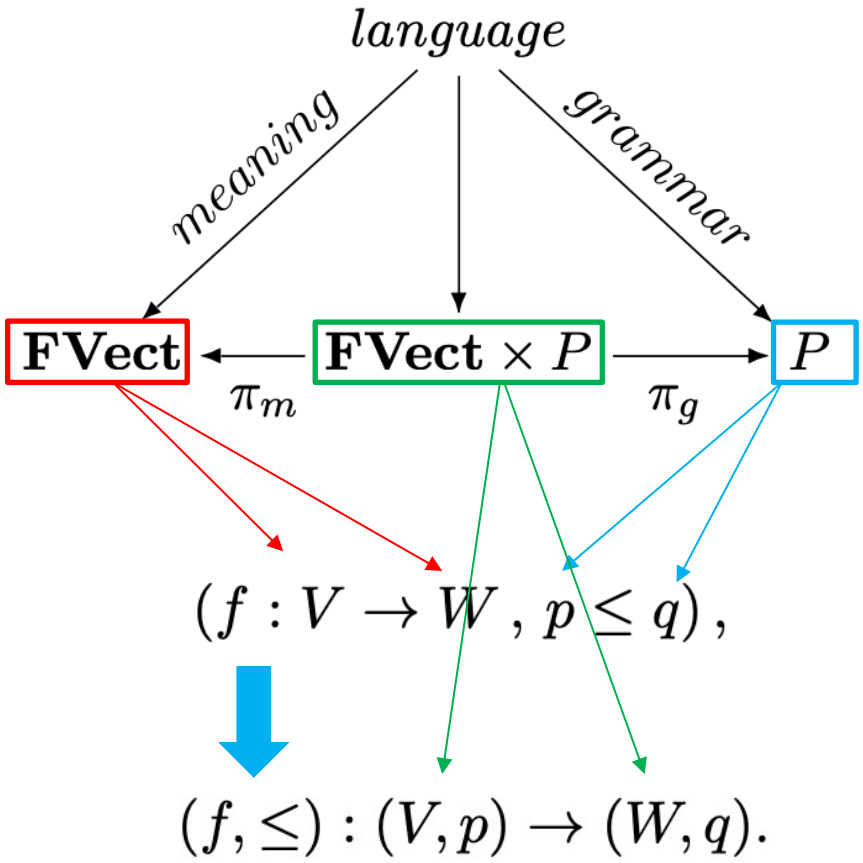
FVect × P 内の射としての「文の意味」

$\alpha = [p_1 \cdots p_n \leq x]$は**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×P** の射である。

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

このfを、「語の意味から文の意味への写像」と呼ぶ。

基本的な射 $(f, \leq)$


language
meaning
grammar
FVect
FVect × P
P
π_m
π_g
$(f : V \to W\ ,\ p \leq q)\ ,$
$(f, \leq) : (V, p) \to (W, q).$

カテゴリー論の応用としての DisCoCat

ここでは、Tai-DanaeによるDisCoCatの解説を紹介する。

Part 3-2

カテゴリー論的構成的分散意味論

カテゴリー論の応用としてのDisCoCat

Tai-Danae Bradley

- カテゴリー論と自然言語処理
- Syntax Category: Pregroup Grammar
- Semantics Category: Vector Space
- Functor: Syntax → Semantics
- “bananas are fruit” の意味を計算する

What is applied category theory

3.2 Natural Language Processing

Tai-Danae Bradley
2018年
https://arxiv.org/pdf/1809.05923.pdf

Monoidal Category

Monoidal カテゴリーは、次の三つ組$(C, \otimes, I)$ である。

- Cはカテゴリー
- $\otimes$は、functor $C \times C \to C$
- I は、$\otimes$について同一性として作用するオブジェクト

全ての要素 A, Bについて$A \otimes B \equiv B \otimes A$の時、Symmetric monoidal カテゴリーという。

Monoidal Categoryの例

- $(\boldsymbol{Set},\ \times,\ \{*\})$
  $\boldsymbol{Set}$ は集合、×は直積、$\{*\}$ は一つの要素だけからなる集合。

- $(\boldsymbol{Top}\ ,\sqcup,\ \emptyset)$
  $\boldsymbol{Top}$は位相空間、⊔は分離和、∅は空集合。

- $(\boldsymbol{FVect},\ \otimes,\ \mathbb{k})$
  $\boldsymbol{Fvect}$は体$\mathbb{k}$上の有限ベクトル空間、⊗はテンソル積、$\mathbb{k}$は体。

compact closed category

その全ての要素が双対 dual を持つmonoidal カテゴリーを compact closed カテゴリーと呼ぶ。

有限ベクトル空間 FVectは、その要素 |v>に対して、双対 <v|を持つので、compact closedカテゴリーである。

compact closed カテゴリーは、ベクトル空間の一般化と考えていい。

ベクトル空間で、その双対 dualの性質を、考えてみよう。

$\mathbb{R}$上のベクトル空間Vでdualを考える

有限次元のベクトル空間Vは、双対空間$V^* = \hom(V, \mathbb{R})$を持つ。
この時、次のような線形写像η_V, ϵ_V を考える。

$$\eta_v : \mathbb{R} \to V \otimes V^*, \qquad \epsilon_V : V^* \otimes V \to \mathbb{R}$$

基底$\{\boldsymbol{e_1}, \cdots \boldsymbol{e_n}\}$を一つ定めると、$V \equiv V^*$ となるので、

$$\eta : \mathbb{R} \to V \otimes V, \qquad \epsilon : V \otimes V \to \mathbb{R}$$

と考えていい。

ベクトル空間のunitとcounit

$\eta: \mathbb{R} \to V \otimes V$をベクトル空間のunitと呼ぶ。
ηは、全ての実数に、$V \otimes V$のベクトルを割り当てる。

$$\eta(1) = \sum_{i,j} c_{ij} \boldsymbol{e_i} \otimes \boldsymbol{e_j}$$

$\epsilon: V \otimes V \to \mathbb{R}$をベクトル空間のcounitと呼ぶ。
ϵは、全ての$V \otimes V$のベクトルに、実数を割り当てる。

$$\epsilon\left(\sum_{i,j} c_{ij} \boldsymbol{v_i} \otimes \boldsymbol{w_j}\right) = \sum_{i,j} c_{ij} (\boldsymbol{v_i} \cdot \boldsymbol{w_j})$$

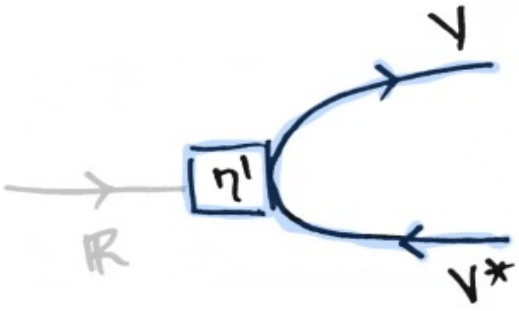
ベクトルv_i, w_jの内積

一般のcompact closed categoryでのunitとcounit

compact closed カテゴリーのオブジェクトVが右双対V^rと左双対V^lを持つとすると、次のようなunitとcounitを考えればいい。

$$\eta_V^l : \mathbb{R} \to V \otimes V^l \qquad \epsilon_V^l : V^l \otimes V \to \mathbb{R}$$
$$\eta_V^r : \mathbb{R} \to V^r \otimes V \qquad \epsilon_V^r : V \otimes V^r \to \mathbb{R}$$

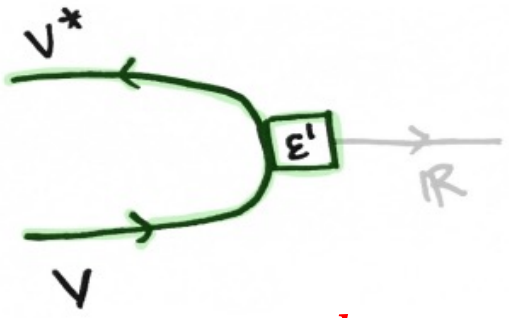
Fvectでは、$V^* = V^r = V^l$になっている、



η^l

$\mathbb{R} \to V \otimes V^l$

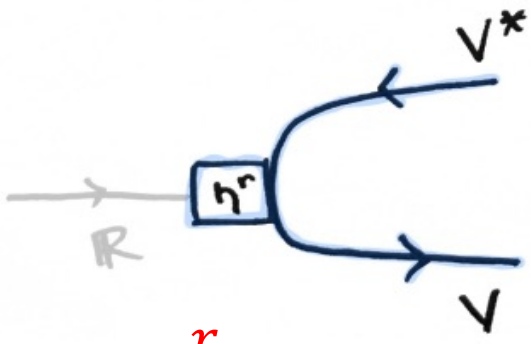
unit



ϵ^l

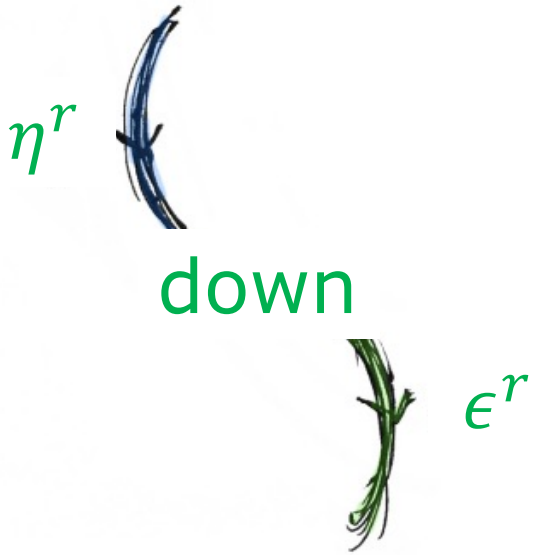
$V^l \otimes V \to \mathbb{R}$

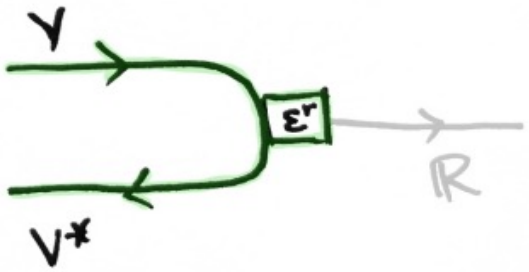
counit



η^r

$\mathbb{R} \to \mathrm{V^r} \otimes V$





ϵ^r

$V \otimes V^r \to \mathbb{R}$

カテゴリー論と自然言語処理

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 → 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。

n ＝ 名詞
s ＝ 文
nn^l ＝ 形容詞
$n^r s n^l$ ＝ 他動詞

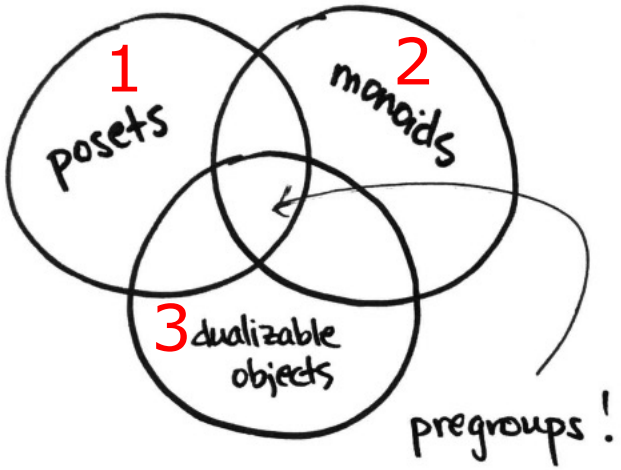
語が連接する時、対応する語の型で、次のような計算を行う。

ww^l ＝ ε^l（消える）
$w^r w$ ＝ ε^r（消える）

Pregroupの定義
Pregroup=poset+monoid+”dual”

1. $(P, \leq)$ は、半順序
2. $p \leq q$ ならば、全ての $a \in P$について、$ap \leq aq$で $pa \leq qa$
3. 全ての $p \in P$ は、左双対 p^l と右双対 p^r を持ち、次の式が成り立つ。 $\epsilon^l \quad \eta^l$ $\epsilon^r \quad \eta^r$

$$p^l p \leq 1 \leq p p^l, \qquad p p^r \leq 1 \leq p^r p$$



pregroupはcompact closed category

先のpregroupの定義中の不等号≥を、カテゴリーの射→と考えればいい。

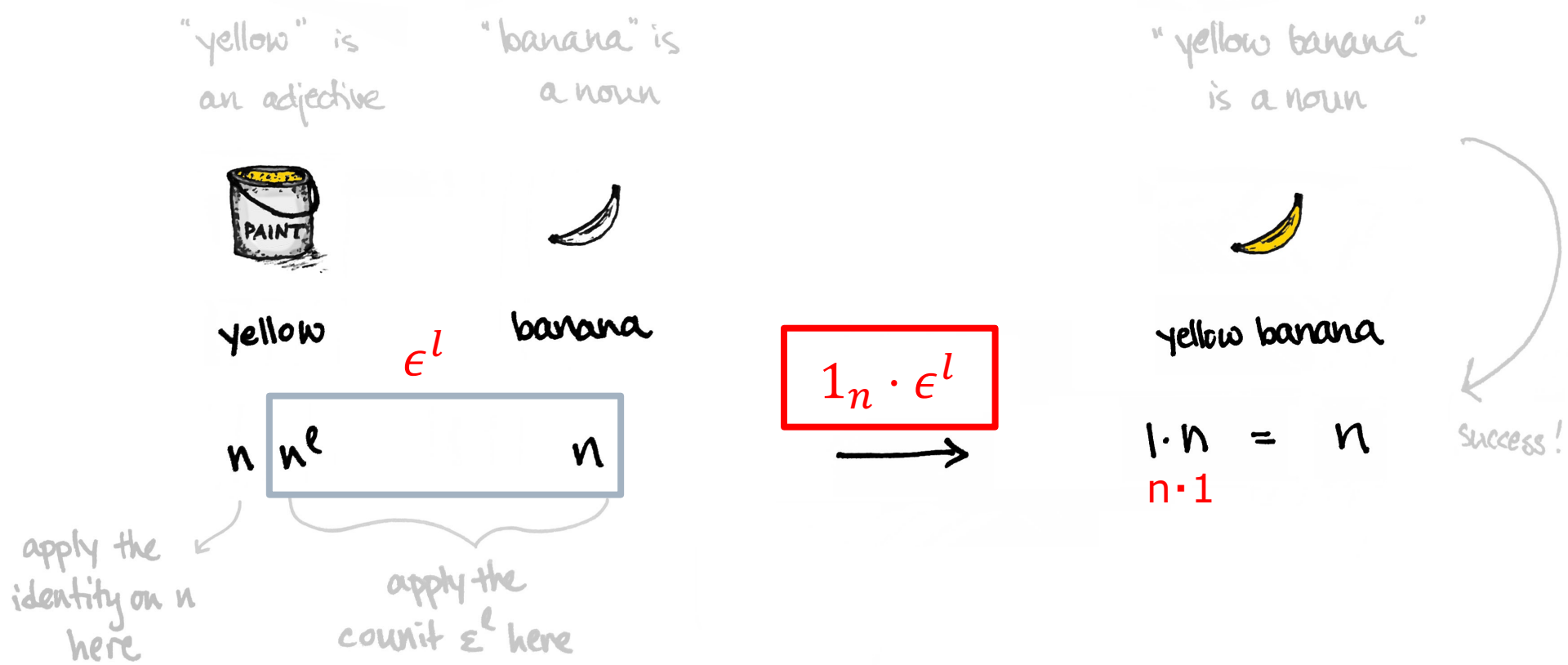
- 射の定義　$p \to q$ iff $p \leq q$
- 射の合成　$p \leq q$ で $q \leq r$ ならp ≤ r
- 同一性　$p \leq p$

pregroupは、次のような半順序以外の性質も持っている。

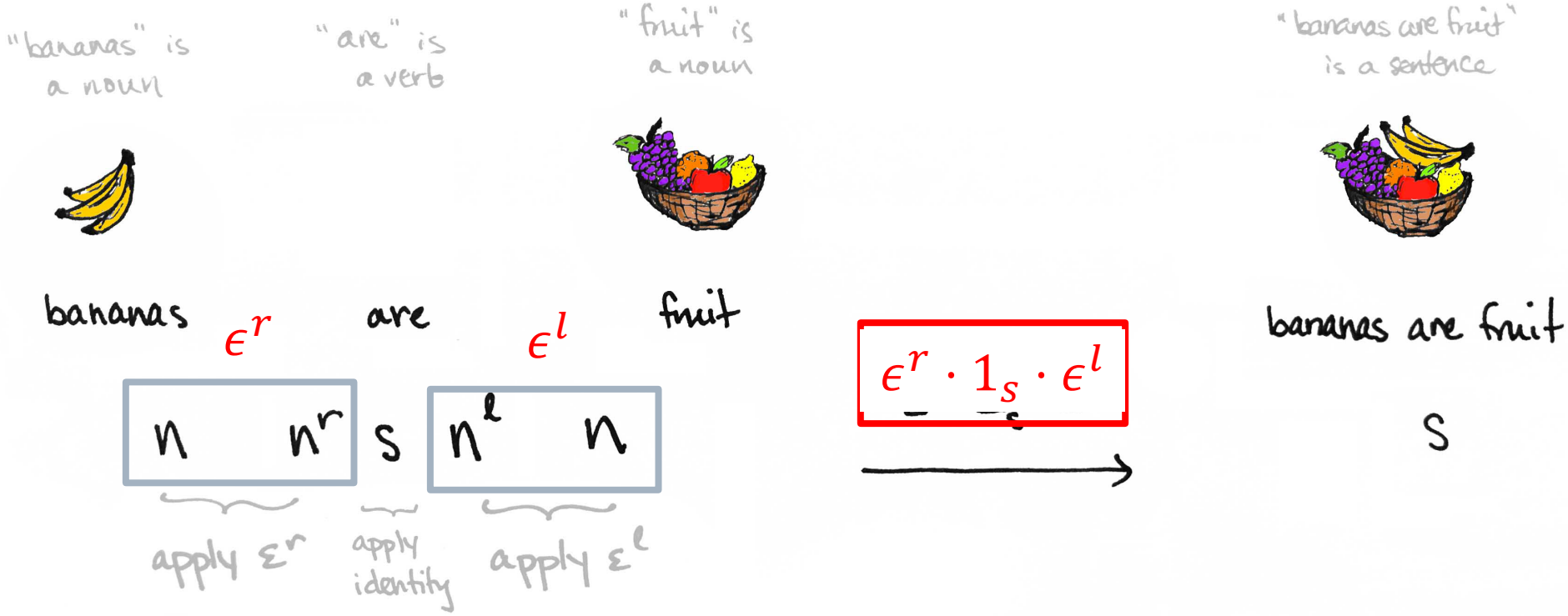
$$p^l p \xrightarrow{\epsilon^l} 1 \xrightarrow{\eta^l} p p^l, \qquad p p^r \xrightarrow{\epsilon^r} 1 \xrightarrow{\eta^r} p^r p$$

pregroupはcompact closed categoryである。

“yellow banana” は名詞として扱われる



“bananas are fruit” は文である



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

The Semantics Category: Vector Spaces = FVec

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。

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

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

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

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

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 → 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。

$$F : PregX \rightarrow FVect$$

ただし、条件がある

$$F : PregX \rightarrow FVect$$

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。別のセッションで説明する。

The Functor: Syntax → Semantics の明示的な記述

この節では、functor F：syntax → semantics の明示的な記述を与える。より具体的には、F：PregX → 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 → 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 → 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: Preg → FVect
Fは、compact closed structure を保存する

- Fは、$Preg\{n,s\}$の**unit** $\eta_n^r: 1 \to n^r n$, $\eta_n^l: 1 \to n n^l$を、
  $FVect$の**unit** $\eta_N: \mathbb{R} \to N \otimes N$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、$Preg\{n,s\}$の**counit** $\epsilon_n^r: n^r n \to 1$, $\epsilon_n^l: n n^l \to 1$を、
  $FVect$の**counit** $\epsilon_N: N \otimes N \to \mathbb{R}$ 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: Preg → FVect
Fは、compact closed structure を保存する

- Fは、$Preg\{n,s\}$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、$Preg\{n,s\}$の複合型を、$FVect$のテンソル積にうつす。

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

“bananas are fruit” の意味を計算する

Step 1: それぞれの語に、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 $N = Fn$ 「文」空間 $S = Fs$ 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

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

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。

$$F\left(n^r s n^l\right) = N \otimes S \otimes N$$

Step 3: 文中の語のベクトル表現を決める

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

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 x 3 の行列として再表現できる。

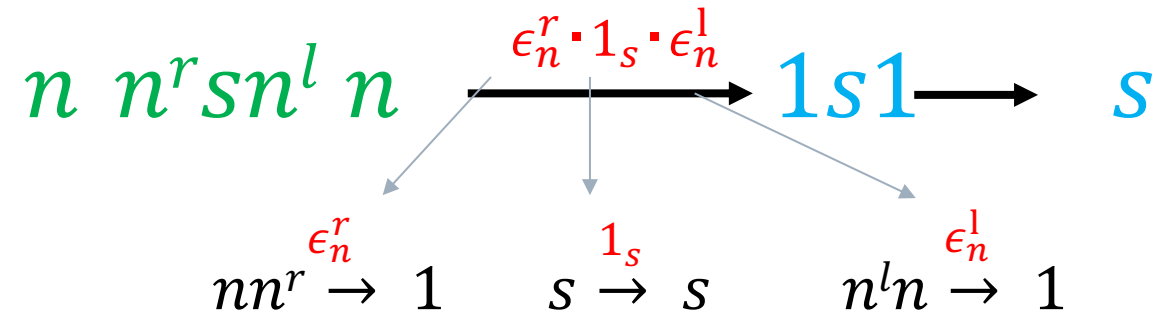
この行列c のi行j列目の係数 c_{ij} は、次のようになる。

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

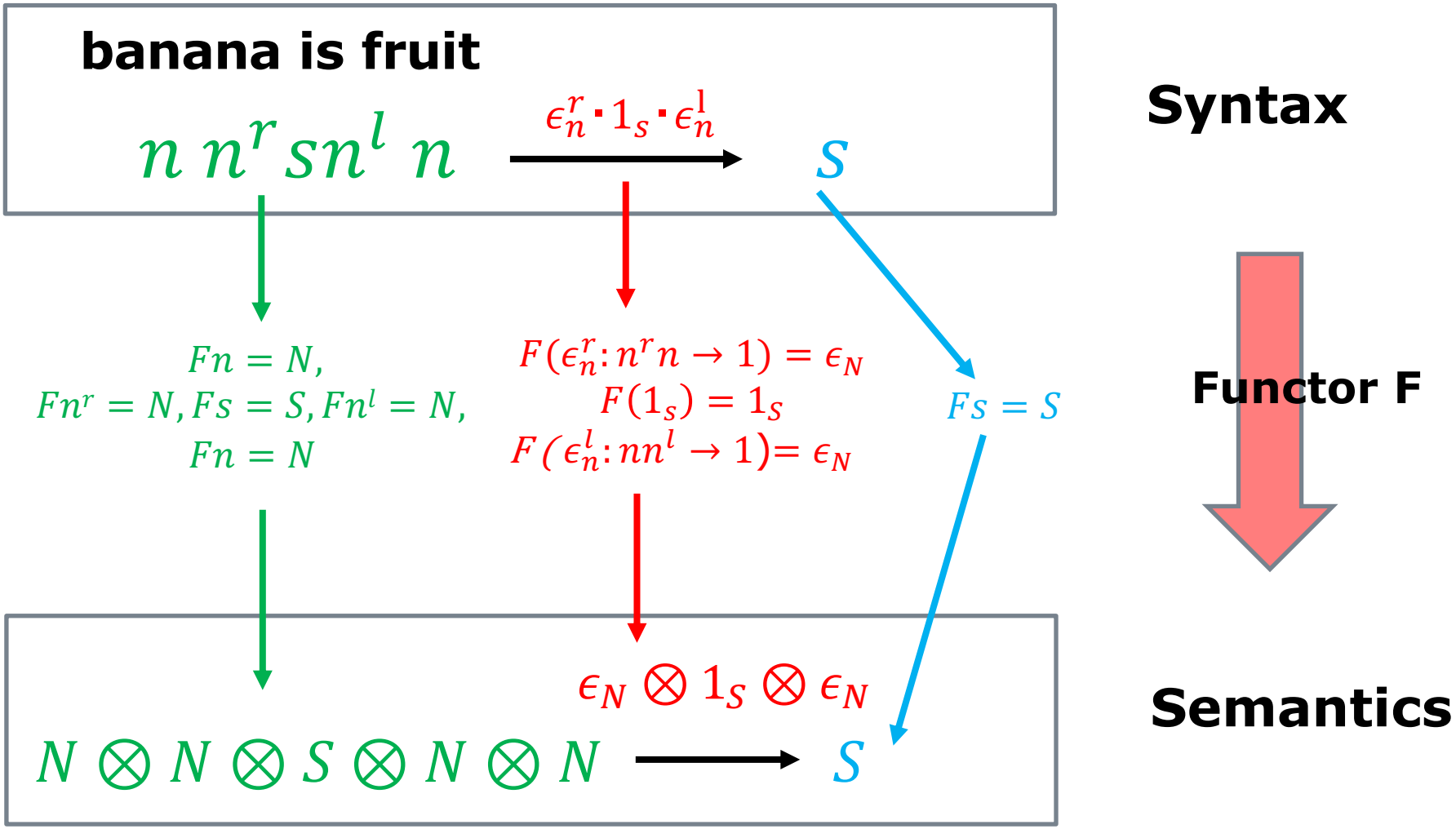
$$\mathbf{are} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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



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、$\epsilon_N: N \otimes N \to \mathbb{R}$という線型写像である。
また、1_Sは、$1_S: S \to S$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、

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

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

Part 4

カテゴリー論的構成的分散意味論の展開

Part 4

カテゴリー論的構成的分散意味論の展開

- Coeckeの Quantum-NLP
- Tai-Danae のReduced Density

Coeckeの Quantum-NLP

CoeckeのQuantum-NLP

このセッションでは、DisCoCatの創始者Bob Coeckeが、現在どのような研究を行っているのかを見てみようと思う。

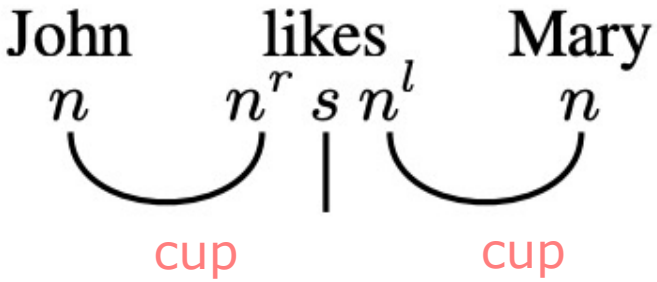
彼は、DisCoCat の枠組みを、ことばの意味を量子状態として捉える方向で発展させ、QNLP 量子論的自然言語処理 Quantum Natural Language Processing を展開した。

実際の量子コンピュータ上でQNLPの実証実験が始まっている。

DisCoCat振り返り

DisCoCat
Pregroup文法での”John likes Mary”の分析

- 語の型　John：n , likes：$n^r s n^l$, Mary：n
- 還元ルール　$p \cdot p^r \to 1$　$p^l \cdot p \to 1$
- ダイアグラム



- 文の型sの導出

$$n \cdot (n^r \cdot s \cdot n^l) \cdot n \to$$
$$(n \cdot n^r) \cdot s \cdot (n^l \cdot n) \to 1 \cdot s \cdot 1 \to s$$

DisCoCat
Functor: pregroupからベクトル空間へ

- Functor $\mathcal{F}$ は、pregroupの型をベクトル空間に移す。

$$n \to N, s \to S, n^r s n^l \to N \otimes S \otimes N$$

  - 他動詞の型 $n^r s n^l$は、次数3のテンソル $N \otimes S \otimes N$ に移る
    これは、$N \otimes N \to S$ の双線形写像と見ることが出来る
  - 形容詞の型$n n^l$は、線形写像$N \to N$を 表わす行列と見ることが出来る

- Functor $\mathcal{F}$ は、pregroupでの還元（reduction）をベクトル空間の縮約(contraction)に移す。

- 文$s = w_1 w_2 \cdots w_n$の意味$\boldsymbol{s}$は、pregroupの還元ルールαの元で、次の式で与えられる。

$$\boldsymbol{s} = \mathcal{F}(\alpha)[\boldsymbol{w_1} \otimes \boldsymbol{w_2} \otimes \cdots \otimes \boldsymbol{w_n}]$$

Foundations for Near-Term Quantum Natural Language Processing

Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, Alexis Toumi

https://arxiv.org/abs/2012.03755v1

2020年

Abstract

近い将来の量子自然言語処理（QNLP）の概念的、数学的基礎を提供し、量子コンピュータ科学者にやさしい言葉でそれを行う。

説明的な表現方法を採用し、実証的な証拠と数学的な一般性についての形式的な記述のための参考文献を提供する。

我々は、我々が採用している自然言語の量子モデルが、言語的な意味と豊かな言語構造（特に文法）を正しく標準的にに結合していることを思い出している。

特に、意味と構造を結合するために量子的なモデルが必要であるという事実は、QNLPが量子系のシミュレーションと同じように、量子ネイティブであることを立証している。

さらに、量子ハードウェア上で古典データをエンコードするための、現在主流のノイズの下での中規模量子（NISQ）パラダイムのさまざまな量子回路は、NISQを特別にQNLPに適したものにしている。

言語構造は、明らかに指数関数的に高い計算コストを必要とするな古典な文法のえンディングとは対照的に、ほとんどただでエンコードすることができる。

図式的な推論はQNLPの中核をなすものである。

まず、量子モデルは、カテゴリー論的量子力学の図式的な形式化によって、言語を量子プロセスとして解釈する。次に、これらの図式はZX-calculusによって量子回路に変換される。そして、意味のパラメータ化が、学習すべき回路変数となる。

我々の言語構造の量子回路へのエンコーディングは、言語構造の中核にウィトゲンシュタインの「意味とは文脈である」と言う議論を据えることによって、現在主流のAI技術を超える、新しい言葉の意味を確立するアプローチを具体化するものである。

自然言語の量子モデル

自然言語の量子モデル

複数の語の意味は、複数のqubitの状態で表現されるとしよう。
特に、hat（帽子）のような名詞 n の意味は、1-qubitの状態 $|\psi_n> \in \mathbb{C}^2$ で表現されるとする。

名詞 n の意味 $\Leftrightarrow 1-qubit$の状態 $|\psi_n> \in \mathbb{C}^2$

名詞 hat の意味 $\Leftrightarrow 1-qubit$の状態 $|\psi_{hat}> \in \mathbb{C}^2$

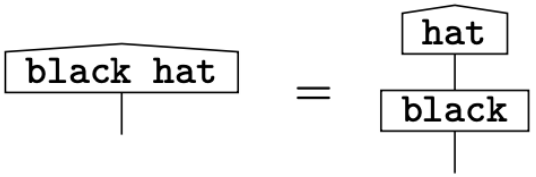
名詞に形容詞を適用する
$hat \rightarrow black\ hat$

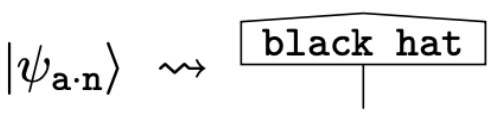
black hatの形容詞 blackは、名詞 hatの状態を black に変える。形容詞aを1-qubitの写像$\eta_a: \mathbb{C}^2 \rightarrow \mathbb{C}^2$ と表わすことが出来る。

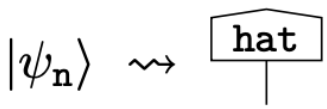
名詞nへの形容詞aの適用の意味$|\psi_{a\cdot n}>$ は次のように表現される。

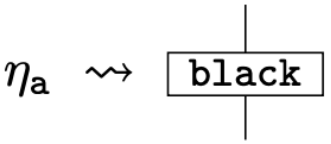
$$|\psi_{a\cdot n}> = \eta_a(|\psi_n>)$$

図式で表すと、次のようになる。
ここで、









名詞に形容詞を適用する (2)
$hat \rightarrow black\ hat$

Blackとhatの意味から black hat の意味を得るもう一つの方法がある。それは、形容詞を 2-qubitの状態$|\psi_a> \in \mathbb{C}^2 \otimes \mathbb{C}^2$と考えることである。

2×2の行列の要素は、次の対応で2-qubitの状態と対応する。

$$\eta_{00}|0\rangle\langle 0| + \eta_{01}|0\rangle\langle 1| + \eta_{10}|1\rangle\langle 0| + \eta_{11}|1\rangle\langle 1| \quad \mapsto \quad \eta_{00}|00\rangle + \eta_{01}|01\rangle + \eta_{10}|10\rangle + \eta_{11}|11\rangle$$

Choi-Jamiolkowski correspondence

この時、名詞に形容詞を適用した時の意味は、次の式で表される。

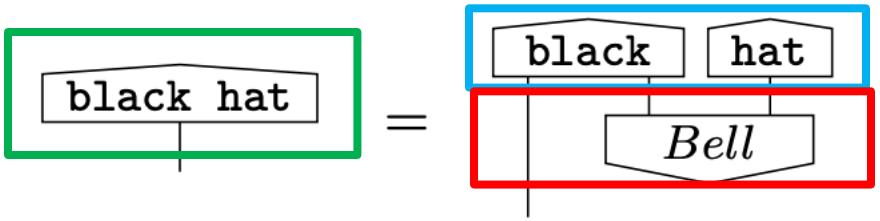
$$|\psi_{a\cdot n}> = (\mathbb{I} \otimes < Bell|\ ) \circ (|\psi_a> \otimes |\psi_n>)$$

ここで$< Bell| = < 00| + < 11|$で、$\mathbb{I}$ は同一性である。

名詞に形容詞を適用する (2)
$hat \to black\ hat$ を図式で表す

$$|\psi_{a\cdot n}> = (\mathbb{I} \otimes < Bell|\ ) \circ (|\psi_a> \otimes |\psi_n>)$$

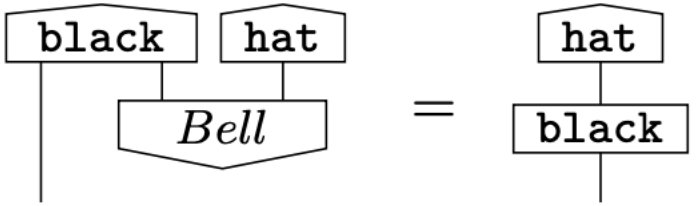
は、図式では、次のように表現される。



形容詞を 2-qubitの状態
$|\psi_a> \in \mathbb{C}^2 \otimes \mathbb{C}^2$と考える

$|\psi_a\rangle \rightsquigarrow$

形容詞aを1-qubitの写像
$\eta_a: \mathbb{C}^2 \to \mathbb{C}^2$ と表わす
$|\psi_{a\cdot n}> = \eta_a(|\psi_n>)$



$|Bell>$ と $<Bell|$
cap と cup

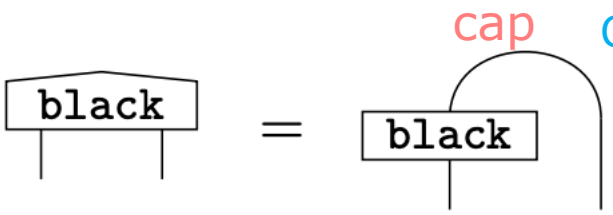
$|Bell>$と$<Bell|$ は、図式的には、次のように表現される。

$|Bell\rangle =$ 

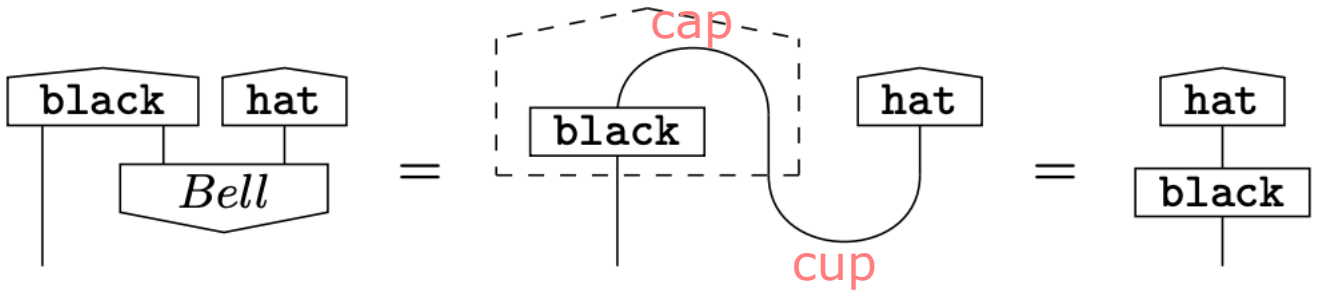
$\langle Bell| =$ 

先に見た、2-qubitの状態を2×2の行列と対応する対応は、次のようになる。

Choi-Jamiolkowski correspondence



この時、



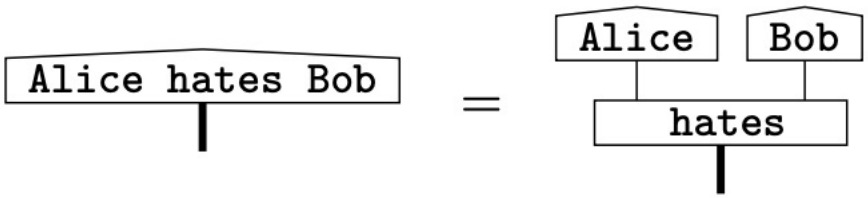
これは、量子テレポーテーションの図式と同じ形をしている。

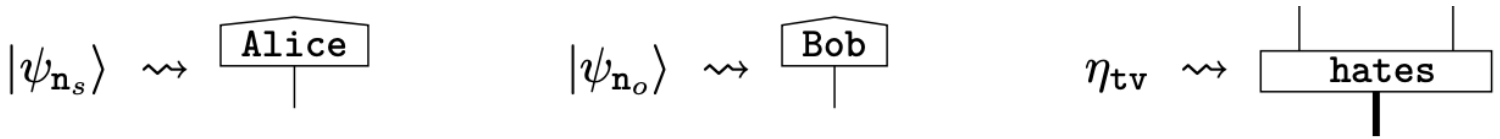
動詞tvに主語n_sと目的語n_oを与える

$|\psi_{n_s}> \in \mathbb{C}^2, |\psi_{n_o}> \in \mathbb{C}^2, |\psi_{n_s \cdot tv \cdot n_o}> \in \mathbb{C}^{2k}$ とする。

kは、文の意味空間のサイズ。

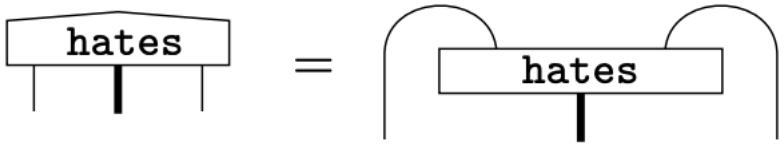
$$|\psi_{n_s \cdot tv \cdot n_o}> = \eta_{tv}(\, |\psi_{n_s}> \otimes\ |\psi_{n_o}>)$$

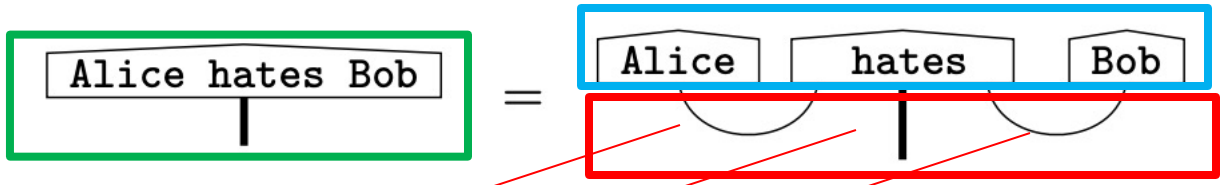




動詞tvに主語n_sと目的語n_oを与える
もう一つのアプローチ

$$|\psi_{tv}> \in \mathbb{C}^2 \otimes (\mathbb{C}^{2k}) \otimes \mathbb{C}^2$$





$$|\psi_{n_s \cdot tv \cdot n_o}> = (< Bell \,|\otimes \mathbb{I} \otimes < Bell|\,) \circ (\,|\psi_{n_s}> \otimes |\psi_{tv}> \otimes |\psi_{n_o}>)$$

QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, Bob Coecke

https://arxiv.org/abs/2102.12846v1

2021年

Abstract

量子自然言語処理（QNLP）は、量子ハードウェア上で実行することを目的とした自然言語処理モデルの設計と実装を扱うものである。本論文では、100文以上のデータセットに対して、Noisy Intermediate-Scale Quantum（NISQ）コンピュータ上で行った最初のNLP実験の結果を紹介する。

Coeckeら(2010)による意味の構成モデルと量子論の形式的な類似性を利用し、量子回路に自然にマッピングされる文の表現を作成する。これらの表現を用いて、量子ハードウェア上で簡単な文の分類タスクを解く2つの自然言語処理モデルの実装と訓練に成功した。

本論文では、これらの実験の主要な原理、プロセス、課題を、自然言語処理研究者が理解しやすい方法で詳細に説明し、実用的な量子自然言語処理への道を開く。

量子コンピュータは、現在の標準的技術より計算速度を指数関数的に向上させるという前提の元で、量子コンピュータ技術はコンピュータサイエンスの最先端分野として急速に発展している。

ただ、最近まで、量子コンピューティングの研究のほとんどは、純粋に理論的であったり、古典的なハードウェア上でのシミュレーションに関心を持っていたのだが、ノイズの下での中間スケール量子（ Noisy Intermediate Scale Quantum ：NISQ）デバイスと呼ばれる研究者が利用できる最初の量子コンピュータの登場により、暗号（Pirandolaら、2020）、化学（Caoら、2019）、生物医学（Caoら、2018）といった幅広いテーマにまたがる有望な実用結果や応用がすでに得られている。

この新しい計算のパラダイムが、自然言語処理にも利用できるのか、という疑問は当然に生まれている。このような応用は、言語関連の問題で計算の高速化を目的とするためだけではない。

計算の高速化を超えて、量子システムがどのように数学的に記述され、また、情報の「量子的」エンコーディングがどのように行われているのかを研究することが、言語の意味の表現と処理の概念的・実用的な進歩につながる可能性がある。

このような観点にインスパイアされた、量子自然言語処理（QNLP）は、まだ始まったばかりの研究分野であり、量子ハードウェア上で実行できるように設計された自然言語処理モデルの開発を目指している。

この分野には素晴らしい理論的研究が存在するが、提案されている実験は古典的なシミュレーションである。例外は最近の研究（Meichanetzidis et al.、2020）で、非常に小規模な概念実証実験が初めて量子ハードウェア上で実行された。

この論文では、量子ハードウェア上で実行される言語処理タスクからなる2つの完全な中規模実験を紹介する。

この実験の目的は、NLPタスクにおいて古典的な実装よりも量子的な優位性を示すことではない。現在利用可能な量子コンピュータの機能が限られているため、このようなことはまだ不可能であると考えている。

本研究では、QNLPが実際にどのようなものであるかについて、NLPコミュニティに詳細な説明を提供することを主な目的としている。従来のモデリングとコーディングのパラダイムがどのように量子に適した形に移行できるかを示し、現在のNISQコンピュータが課す課題と制限を探る。

Bob Coecke

CoeckeのOxfordでの担当コースシラバス
Classical and Quantum Compositional Distributional Meaning (2022-23)

自然言語（コンピュータとは異なる）の意味をモデル化することは、人工知能において最も困難な問題の一つである。この問題を解決することで、テキストの要約、検索、機械翻訳、言語生成、質問応答など、さまざまなテキストや言語処理アプリケーションの品質やインパクトを劇的に向上させる可能性がある。

この問題に対しては、長年にわたって多くの異なるアプローチが考案されてきた。その一つが「形式的意味論」で、自然言語を高次の論理に「コンパイル」するプログラミング言語として扱う。また、「分散意味論」は、言葉の意味を、出現文脈によって決まる高次元の意味空間の点としてモデル化するものである。

https://courses.maths.ox.ac.uk/course/view.php?id=833

CoeckeのOxfordでの担当コースシラバス
Classical and Quantum Compositional Distributional Meaning (2022-23)

最近の研究では、この2つのアプローチの長所を調和させ、意味の構成的な分散型（ベクトルベース）モデルを作り出すという課題に取り組んでいる。

このコースは、この新しく急速に成長している分野の理論的な端緒となるもので、フレーズやセンテンスの意味を形成するために単語の意味をどのように構成するかに焦点を当てている。

特に、量子自然言語処理（QNLP）という新しい分野は、現在DisCoCatとして知られているこの言語モデルに基本的な方法で依存している。QNLPは、量子コンピュータの最も有望な手段の1つと考えられている。

https://courses.maths.ox.ac.uk/course/view.php?id=833

Tai-Danaeの reduced density

最初におわび

今回のセミナーでは、Tai-Danae Bradleyの次の論文の紹介を中心にする予定でした。

- Language Modeling with Reduced Densities
  https://arxiv.org/abs/2007.03834v4

ただ、今回は他の部分が膨らんで、また、準備に十分な時間が取れず断念しました。すみません。

この論文の数学的基礎については、2023年2月のマルレク「密度行列 ρ で理解する確率の世界」がその解説になっています。https://www.marulabo.net/docs/density2/

先日、講演資料と講演ビデオを公開したので、そちらを参照ください。

Language Modeling with Reduced Densities

Tai-Danae Bradley

2021年

https://arxiv.org/abs/2007.03834v4

なぜ、Tai-Danaeの議論に注目するのか

なぜ、DisCoCatの流れの中で、Tai-Danae の議論に注目するのでしょうか？この論文の冒頭で、彼女はこう言っています。

「この研究は、今日の最先端の統計的言語モデルが、その性能において印象的であるだけでなく、より本質的に重要なことは、それが構造化されていないテキストデータの相関関係から完全に構築されているという観察から生まれたものです。」

彼女の関心は、現在の「大規模言語モデル」の「印象的」な成功に向けられています。彼女はそれがDisCoCatモデルとは少し異なる言語モデルであることも知っています。その上で、その背後にあるものを探り出そうとしているのです。

問題の立て方が重要であること

僕にとって印象的だったのは、彼女が次々と問題を立てることでした。答えの前には、もちろん、問題があります。ただ、答えを見つける条件が成熟するというのは、正しく問題をたてることができるということです。

- 自然言語における表現の意味をとらえる数学的構造は何か？
- この構造は、テキスト・コーパスを用いてどの程度まで十分に検出できるのか？
- 抽象的な概念とその相互関係を自然に掘りだす方法はあるのか？
- 論理と命題の連関はどのようにして生まれるのか？

二つの基本的問題

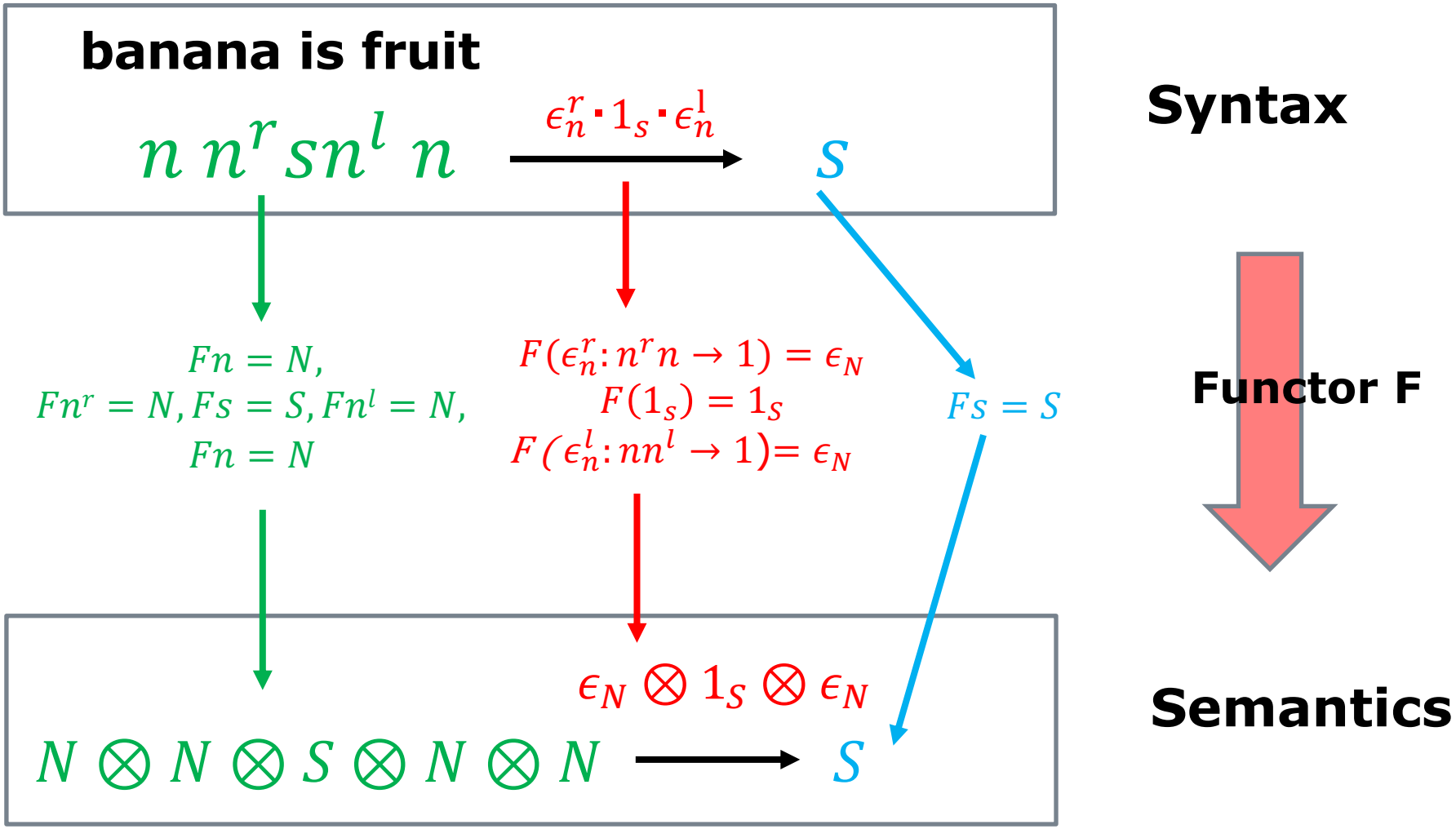
こうして、彼女は、次の二つが基本的な問題だとします。

- どのような数学的構造が、構造化されていないテキストデータには存在するのか？
- どのようにして、テキストの情報は、カテゴリー構造を維持したまま保存され、モデル化することができるのか。

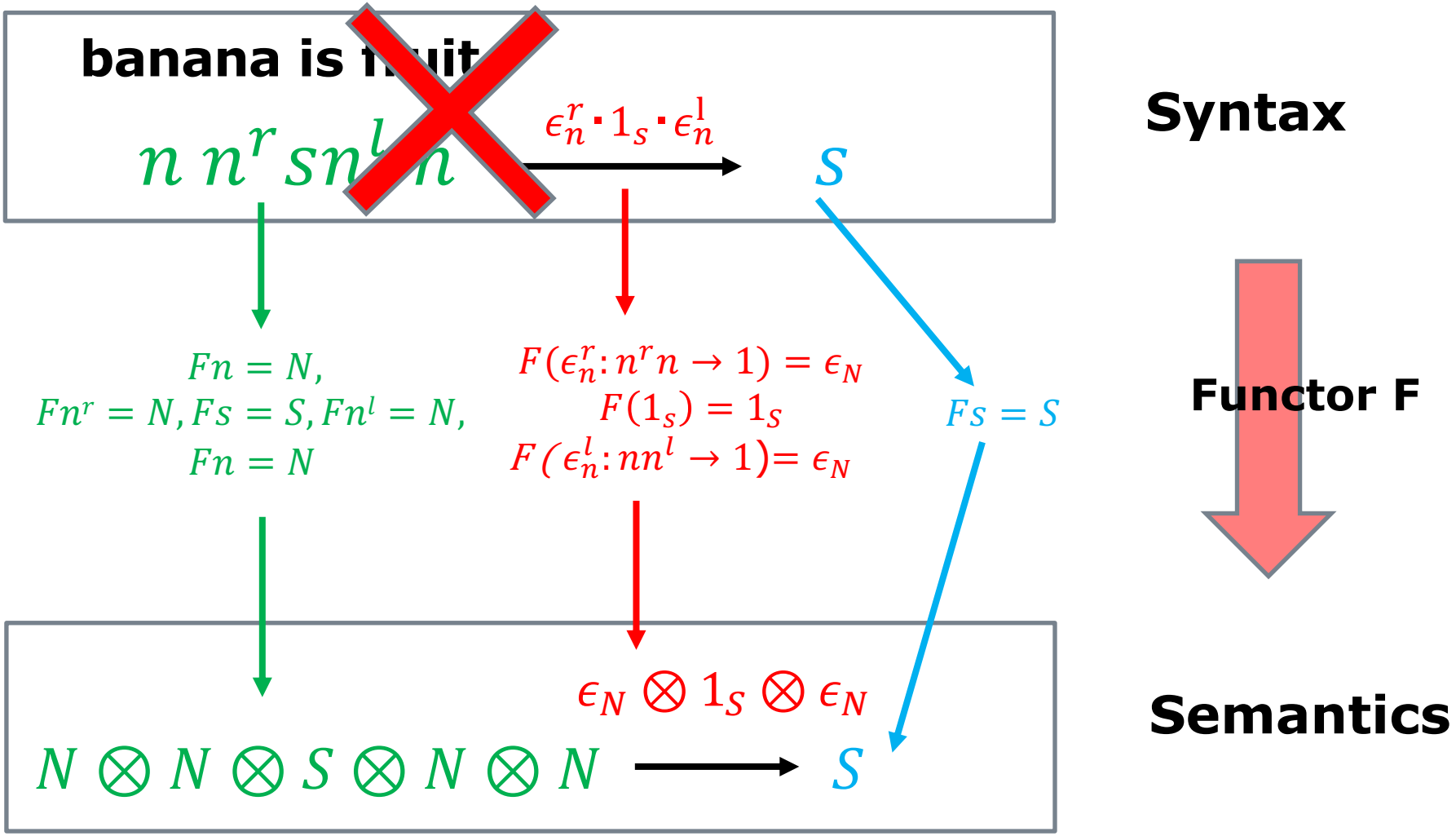
先の論文は、この問題に対する彼女の答えを述べたものです。

彼女の答えは数学的に表現されたものなのですが、すくなくとも彼女の「問題意識」は、わかりやすいものだと思います。

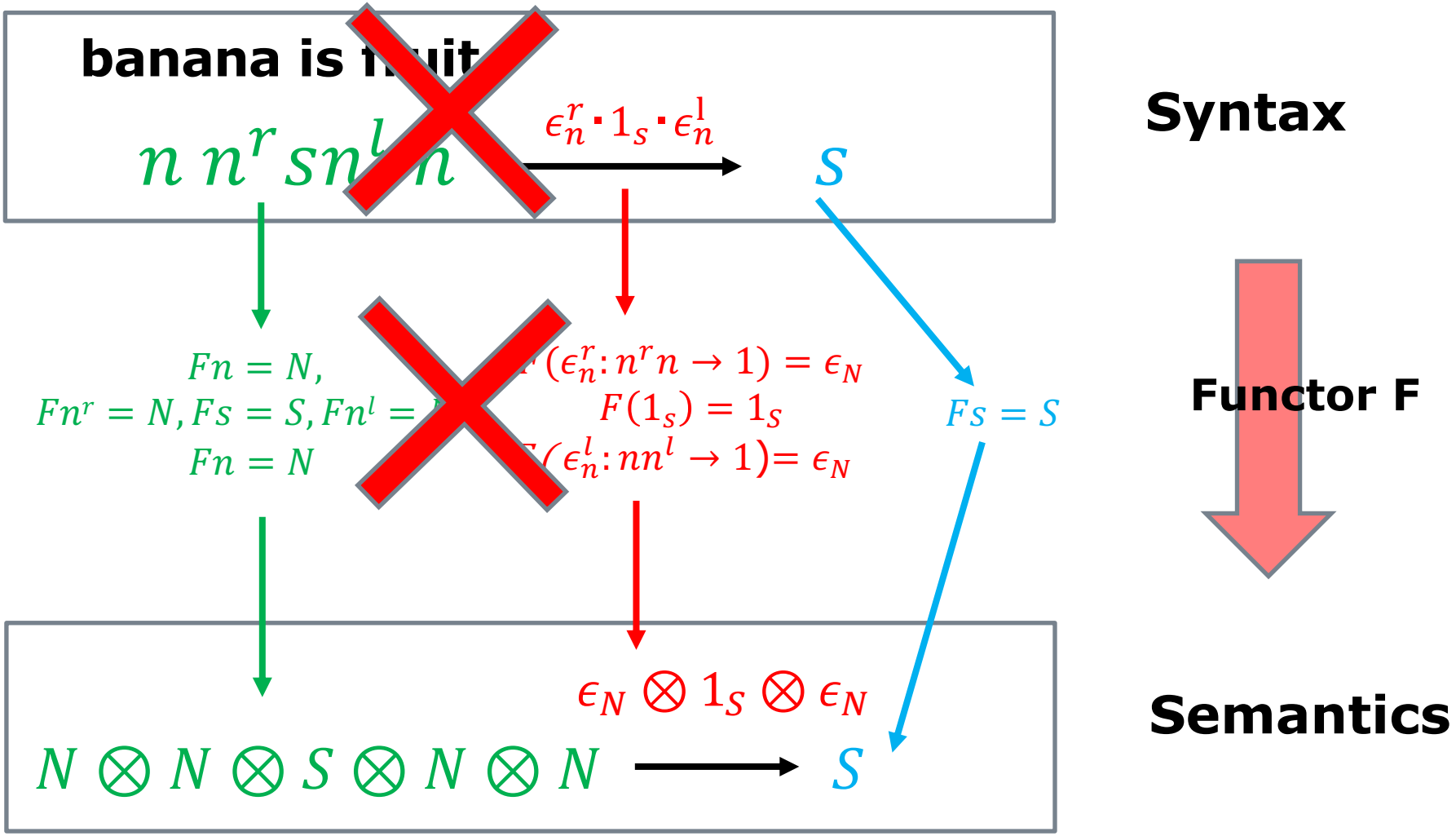
以前の彼女の図式


banana is fruit
$n\ n^r s n^l\ n$
$\epsilon_n^r \cdot 1_S \cdot \epsilon_n^l$
S
$Fn = N,$
$Fn^r = N, Fs = S, Fn^l = N,$
$Fn = N$
$F(\epsilon_n^r : n^r n \to 1) = \epsilon_N$
$F(1_s) = 1_S$
$F(\epsilon_n^l : n n^l \to 1) = \epsilon_N$
$Fs = S$
$\epsilon_N \otimes 1_S \otimes \epsilon_N$
$N \otimes N \otimes S \otimes N \otimes N$
S
Syntax
Functor F
Semantics

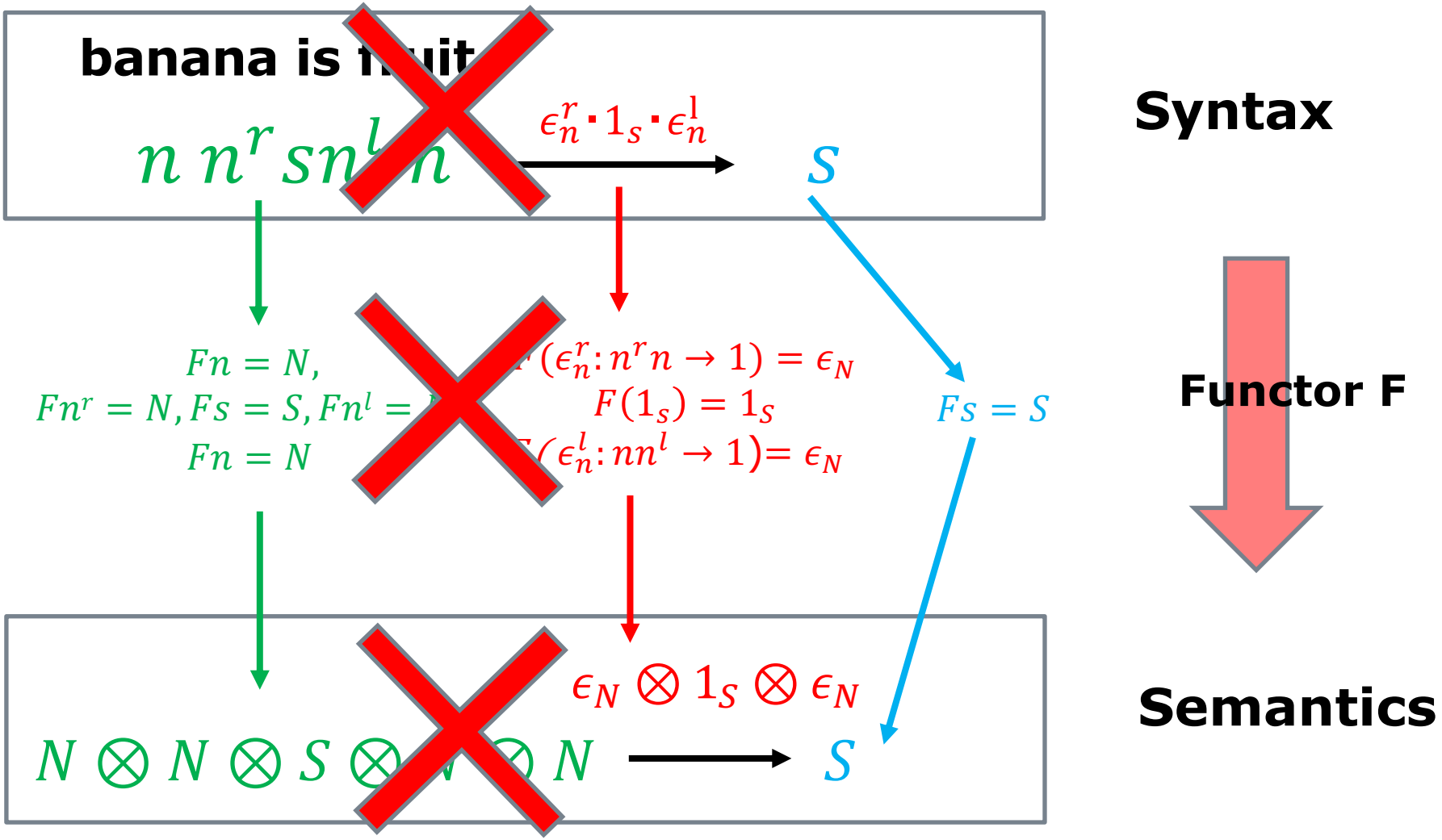
構造化されていないテキスト



構造化されていないテキスト



構造化されていないテキスト



構造化されていないテキスト