

自然言語処理入門 「黒い目の大きな女の子」

東京大学 情報基盤センター

(情報理工学系研究科数理情報学専攻、学際情報学府 兼担)

中川裕志

nakagawa@dl.itc.u-tokyo.ac.jp

<http://www.r.dl.itc.u-tokyo.ac.jp/~nakagawa/>

参考文献

- ✓ 岩波講座 言語の科学 全11巻
 - ✓ 形態素解析は第3巻、構文解析は第8巻、談話、対話は第7巻、文法と意味は第4巻、音声は第2巻、情報処理は第9巻
- ✓ 東大出版会 言語と計算
 - ✓ 談話、対話は第3巻、統計的言語処理は第4巻、情報検索は第5巻
- ✓ 学会誌、国際会議論文集など
 - ✓ 言語処理学会誌「自然言語処理」(中川は編集長をしています。)
 - ✓ 人工知能学会誌
 - ✓ 情報処理学会 論文誌
 - ✓ Computational Linguistics (ACL の journal)
 - ✓ Proceedings of ACL
 - ✓ Proceedings of COLING
 - ✓ ACM SIGIR

黒い目の大きな女の子

- 形態素に解析できると次は、形態素間の関係を知りたい
- 1文における形態素間の関係を構文と言い、これを調べることを構文解析という
- 方法1: 句構造を明らかにする
- 方法1: 係り受け解析
 - 日本語での係り受け解析は構文解析の一種
- 「黒い目の大きな女の子」ではいったいいくつくらいの係り受け構造が可能だろうか？
 - なお、日本語の場合、係り受け関係が交差しないという条件がある。

句構造文法と書き換え規則

■ 文法とは、

- 語彙 (終端記号)

- 文法範疇 (非終端記号、品詞)

- 書き換え規則

- 開始記号 (文)

■ の4要素からなる。

- 文の生成は、開始記号から始めて書き換え規則で書き換えて全ての文法範疇が語彙に置き換える

- 文の構文解析は、文の要素 (形態素) を語彙にマッチさせ、語彙から書き換え規則によって文法範疇に書き換え、最後に開始記号にたどり着く計算過程である。

文法の例

- ◆語彙： 太郎、花子、次郎、が、を、と、殴る
- ◆文法範疇：名詞、動詞、助詞、名詞句、後置詞句、動詞句、文
- ◆開始記号：文
- ◆書き換え規則
 - ◆名詞→太郎、名詞→花子、名詞→次郎、助詞→が、
助詞→を、助詞→と、助詞→は、動詞→殴る
 - ◆後置詞句→名詞・助詞、後置詞句→名詞句・助詞、
名詞句→名詞、動詞句→動詞、
動詞句→後置詞句・動詞句、動詞句→文、
名詞句→動詞句・名詞句

1. 文→後置詞句・動詞句
2. 文→名詞・助詞・動詞句
3. 文→太郎 が・動詞句
4. 文→太郎 が・後置詞句・動詞句
5. 文→太郎 が・名詞・助詞・動詞句
6. 文→太郎 が 次郎 を・動詞
7. 文→太郎 が 次郎 を 殴る

(補遺)形式言語理論とオートマトン

- ◆ここまで使ってきた文法は文脈自由文法と呼ばれるものである。
- ◆この他に、3種類の文法のカテゴリーがある。 X, Y, Z を文法範疇、 a, b を単語とすると
 - ◆正規文法: $X \rightarrow aY, X \rightarrow a$
 - ◆文脈自由文法: $X \rightarrow YZ, X \rightarrow a$
 - ◆文脈依存文法: 文脈自由文法に加えて $aXb \rightarrow aYb$ も許す。ただし、 Y は X より長い。
 - ◆0型文法: 文脈依存文法に加え、 $XY \rightarrow Z$ というような書き換え結果が短くなる規則も許す
- ◆このような分類を考えたのがChomsky。Chomsky階層

つづき

- ◆これらの文法を解析できる機械も分かっている。
- ◆正規文法： **有限状態オートマトン**。記憶装置のない回路のようなもの
- ◆文脈自由文法： **プッシュダウンオートマトン**。先入れ後出しの記憶のある回路
- ◆文脈依存文法： **線形有界オートマトン**。書き込みと消去可能な有限長のテープを持つ回路。ほぼ現代の計算機
- ◆0型言語： **チューリングマシン**。テープ長が無限の回路。実質的に現在の計算機。

補遺はここまで

一筋縄ではいかない例

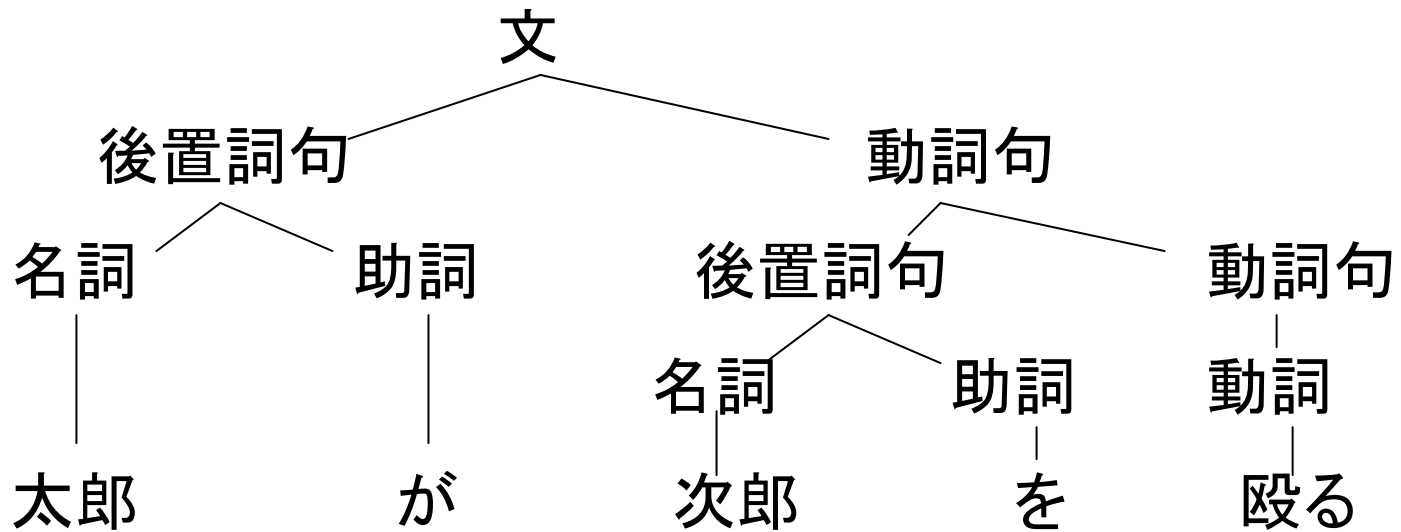
- 次郎は太郎が買った本を読んだ
- 太郎が次郎と花子を殴る
- I saw a girl with a telescope.
- Since Jay always walks a mile seems like a short distance to him.

構文構造の表現

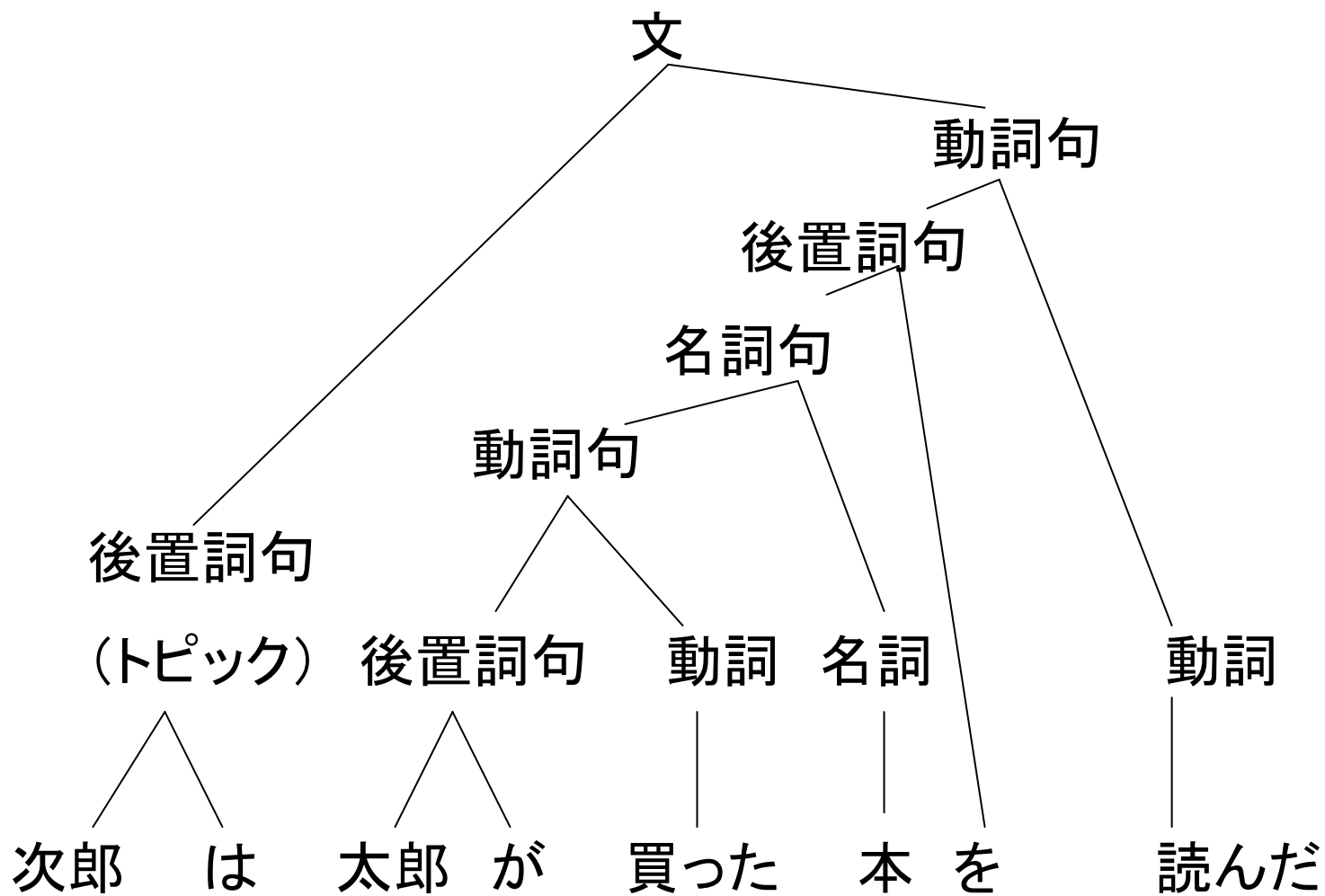
- 括弧

- (太郎が((次郎を)(殴る)))

- 構文木

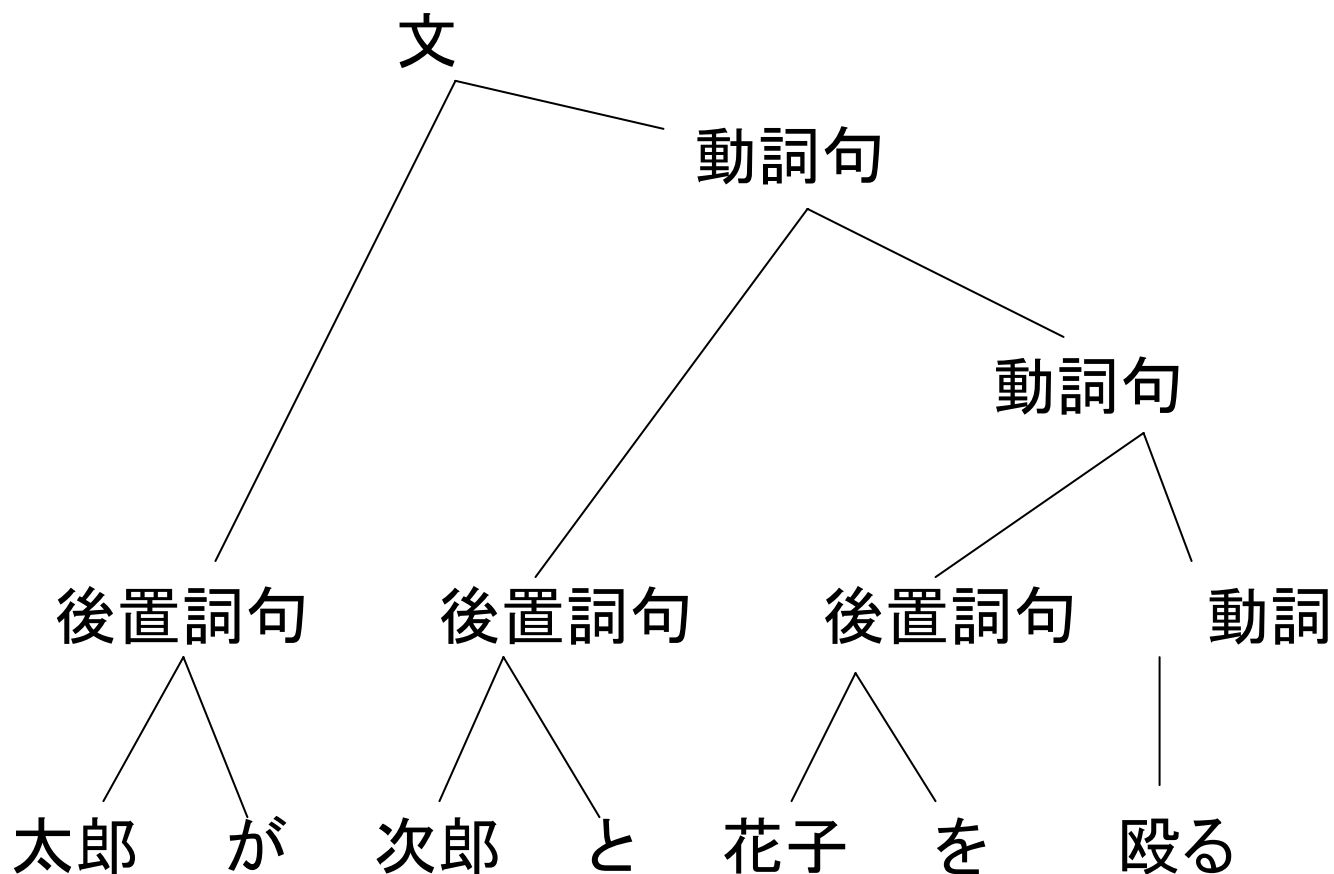


■次郎は太郎が買った本を読んだ



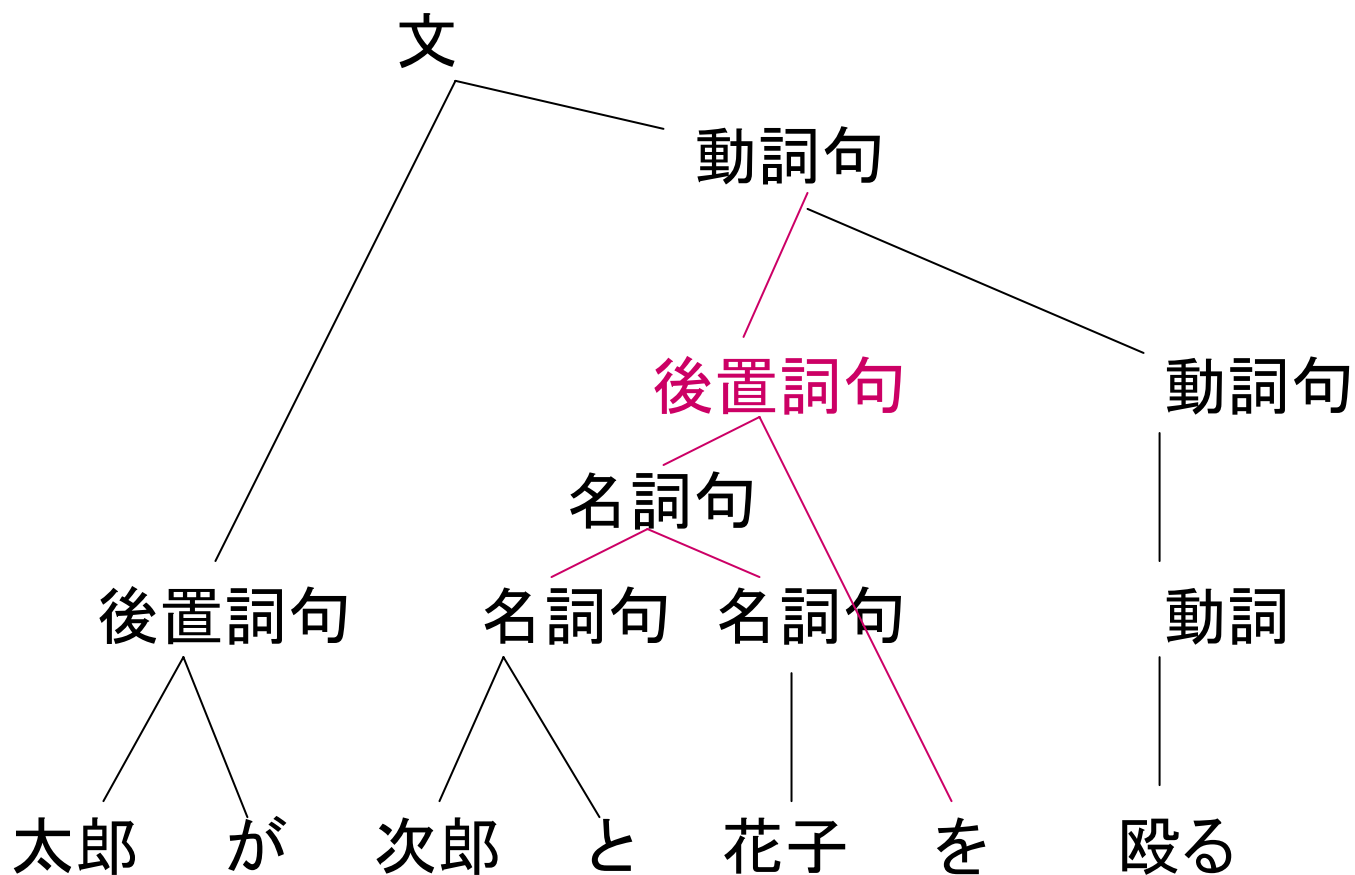
■太郎が次郎と花子を殴る

名詞句→名詞 と 名詞、



■太郎が次郎と花子を殴る

名詞句→名詞 と 名詞、



一筋縄ではいかない例

- 次郎は太郎が買った本を読んだ
- 太郎が次郎と花子を殴る
- I saw a girl with a telescope.
- Since Jay always walks a mile seems like a short distance to him. → 袋小路文 (garden path sentence)

構文解析アルゴリズム

- 構文解析は自然言語処理の重要な部分ではあるが全てではない
- 一時期、自然言語処理の研究といえば**構文解析**という感じであった。
- 星の数ほど構文解析アルゴリズムがある。全部は紹介しきれないし、必要もない。
- 分類
 - 下降型、上昇型、左隅型
- 有力なアルゴリズムをひとつ紹介しておくことにする。

下降型構文解析アルゴリズム

例とする文法

- ◇ $S \rightarrow VP$, $VP \rightarrow \text{できる}$, $VP \rightarrow PP V$, $VP \rightarrow Adv VP$,
 - ◇ $Adv \rightarrow \text{すぐ}$, $V \rightarrow \text{ほめる}$, $PP \rightarrow NP \text{を}$, $NP \rightarrow \text{太郎}$, $NP \rightarrow VP NP$
 - ◇ 0 すぐ 1 できる 2 太郎 3 を 4 ほめる 5
 - ◇ ここで、0から5を位置という。
-
- ◇ 文法範疇: 名詞:N、動詞:V、名詞句:NP、動詞句:VP、副詞:Adv、文:S など

下降型構文解析アルゴリズム

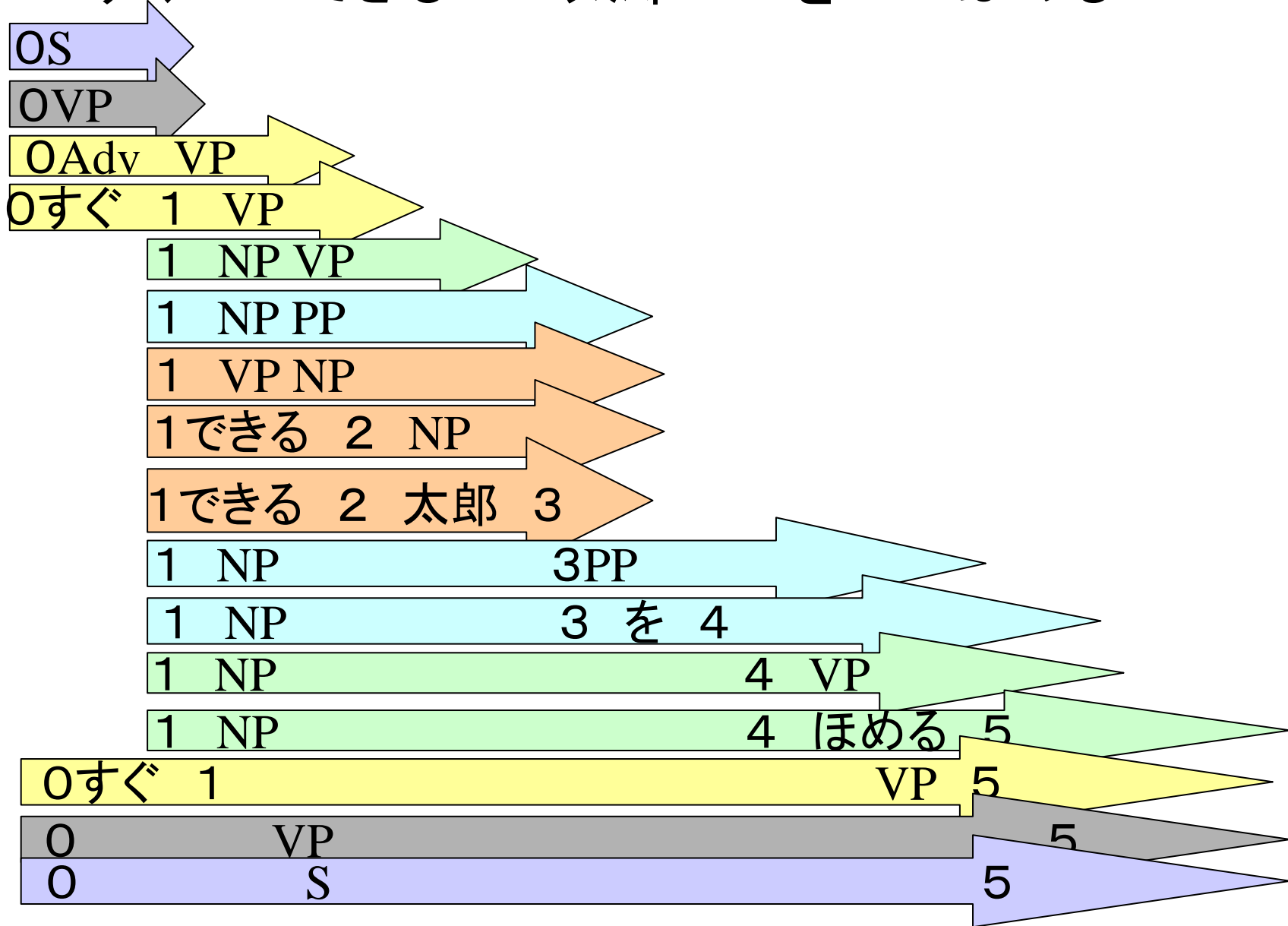
- ✧ $S \rightarrow VP$, $VP \rightarrow$ できる, $VP \rightarrow PP V$, $VP \rightarrow Adv VP$,
- ✧ $Adv \rightarrow$ すぐ, $V \rightarrow$ ほめる, $PP \rightarrow NP$ を, $NP \rightarrow$ 太郎, $NP \rightarrow VP NP$
- ✧ 0 すぐ 1 できる 2 太郎 3 を 4 ほめる 5
- ✧ ここで、0から5を位置という。
- ✧ 位置*i* から 位置*j*を解析した結果*X*という文法範疇(品詞、名詞句、動詞句、節など)になったことを具現化(instantiation)といい、 $X(i, j)$ と書く。*X*が位置*i*以降の解析結果となることを $X(i)$ と書く。
- ✧ 書き換え規則の右辺の各文法範疇の具現化も同様で、
- ✧ $VP \rightarrow 1PP V$, $VP \rightarrow 1PP2V$, $VP \rightarrow 1PP2V3$ などと書く。

解析アルゴリズムの概要

1. 入力文の場所 $i = 0$ とする。 $X(i) = \text{文}$ とする
 2. $X(i)$ をスタックに push する
 3. $X \rightarrow a$ の形の規則を探す。単語 a が入力文の i から $i+1$ に一致すると $X(i, i+1)$ と具現化し、スタックから pop して除く。
 $i = i + 1$
 4. if 3. で規則がない then $X \rightarrow YZ$ の形の規則を探し、結果を $R = \{X \rightarrow YZ\}$ (=規則の集合) とする
 5. foreach (R)
 { $Y(i)Z$ をスタックに push する
 3. 以下を再帰的に実行 }
 6. 入力文末まで読み、
 if スタックが空 then 解析成功 else 失敗
5. の foreach の再帰処理で全ての可能な規則を適用して、可能な全解析を生成

解析例

- 0 すぐ 1 できる 2 太郎 3 を 4 ほめる 5



□ 入力文は「すぐできる太郎をほめる」

□ 入力単語

スタック

具現化規則

- | | | | |
|----|----------|---------------------------------------|--------------|
| 1. | | S(0) | |
| 2. | | VP(0)S(0) | S → 0VP |
| 3. | | Adv(0)VPVP(0)S(0) | VP → 0AdvVP |
| 4. | すぐ(0,1) | Adv(0,1)VPVP(0)S(0) | Adv → 0すぐ1 |
| 5. | | VP(1)VP(0)S(0) | VP → 0Adv1VP |
| 6. | | PP(1)V VP(1)VP(0)S(0) | VP → 1PP V |
| 7. | | NP(1)PP PP(1)VVP(1)VP(0)S(0) | PP → 1NP PP |
| 8. | | VP(1)NPNP(1)PP PP(1)VVP(1)VP(0)S(0) | NP → 1VP NP |
| 9. | できる(1,2) | VP(1,2)NPNP(1)PP PP(1)VVP(1)VP(0)S(0) | VP → 1できる2 |

- | | | | |
|-----|----------|-------------------------------------|-------------|
| 1. | | NP(2)NP(1)PP PP(1)VVP(1)VP(0)S(0) | NP→1 VP2NP |
| 2. | 太郎(2,3) | NP(2,3)NP(1)PP PP(1)VVP(1)VP(0)S(0) | NP→2太郎3 |
| 3. | | NP(1,3)PP PP(1)VVP(1)VP(0)S(0) | NP→1 VP2NP3 |
| 4. | | PP(3)PP(1)VVP(1)VP(0)S(0) | PP→1NP3を |
| 5. | を(3,4) | PP(1,4)VVP(1)VP(0)S(0) | PP→1NP3を4 |
| 6. | | V(4)VP(1)VP(0)S(0) | VP→1PP4V |
| 7. | ほめる(4,5) | V(4,5)VP(1)VP(0)S(0) | V→4ほめる5 |
| 8. | | VP(1,5)VP(0)S(0) | VP→1PP4VP5 |
| 9. | | VP(0,5)S(0) | VP→0Adv1VP5 |
| 10. | | S(0,5) | S→0VP5 |

□結局、スタックにおいて $X(i) \rightarrow X(i,j)$ という書き換えをしている。

□適用できる書き換え規則の種類が複数あるときはそれぞれに応じて別々の解析を進める。

□上の解析例では「すぐ」が「ほめる」にかかったが、

□例文解析の1枚目の3. で $VP \rightarrow Adv VP$ という規則の代わりに

$VP \rightarrow PP VP$ という規則を適用すると「すぐ」が「できる」にかかる構造になる。

□このような解析をしながら同時に構文木も生成する。構文木によってかかり受け関係が明らかになる。

係り受け解析器

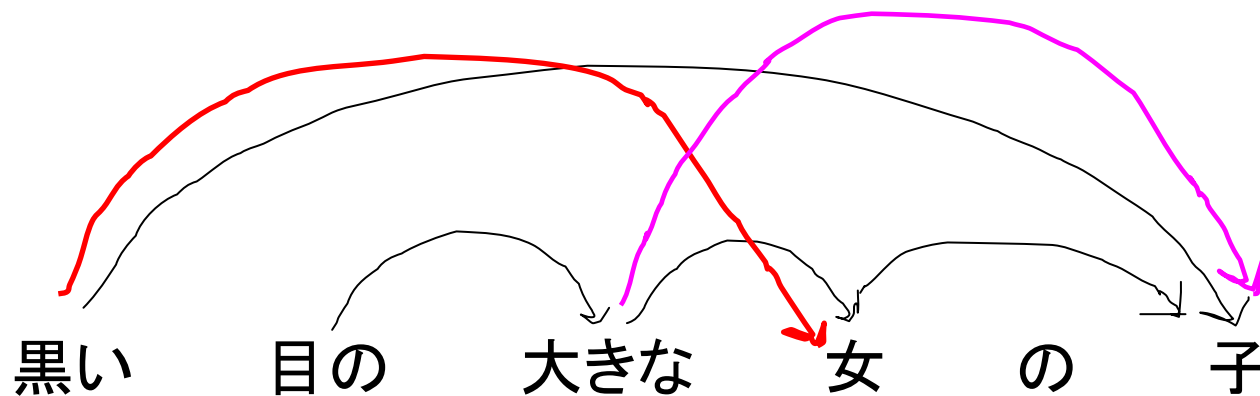
- 日本語文の係り受け解析を行うシステムでは、形態素解析システムの解析結果(形態素列)を入力とし、それらを文節単位にまとめ、文節間の係り受け関係を決定します。
- 文節と文節の間に係り受け関係の有無がある。
- したがって、与えられた文字列を文節にまとめるところから開始。
- 係り受けを制限するのは非交差条件。

係り受けの制約

- 係り受け解析では、係り受けの非交差が基本的制約になる。

非交差

交差



係り受け可能性の規則の例

- 文節が以下の条件のいずれかを満たし
 - (係:デ格 読点) (係:カラ格 読点) (係:マデ格 読点))
- かつ、その後方の用言が以下の条件を持つ
 - (用言の係られる度合い:強)

ならば

- 通常の係り受け関係で係る

➤ このような規則をありとあらゆる種類の文節に対して定義しておく。

係り受け解析の動作

1. 同形異義語を処理し、一意の形態素列に変換
2. 各形態素にたいして、その働きを示す種々のマークを与える。これは辞書的情報、文節にまとめるための自立語・付属語の例外的情報など
3. 形態素に与えられたマークにしたがって、形態素列を文節列に変換
4. 各文節に対して、その働きを示すマークを与える。用言、体言、ガ格、ヲ格、並列構造の可能性など。

なお専門的には上記のマークをfeatureと呼ぶ

係り受け解析の動作

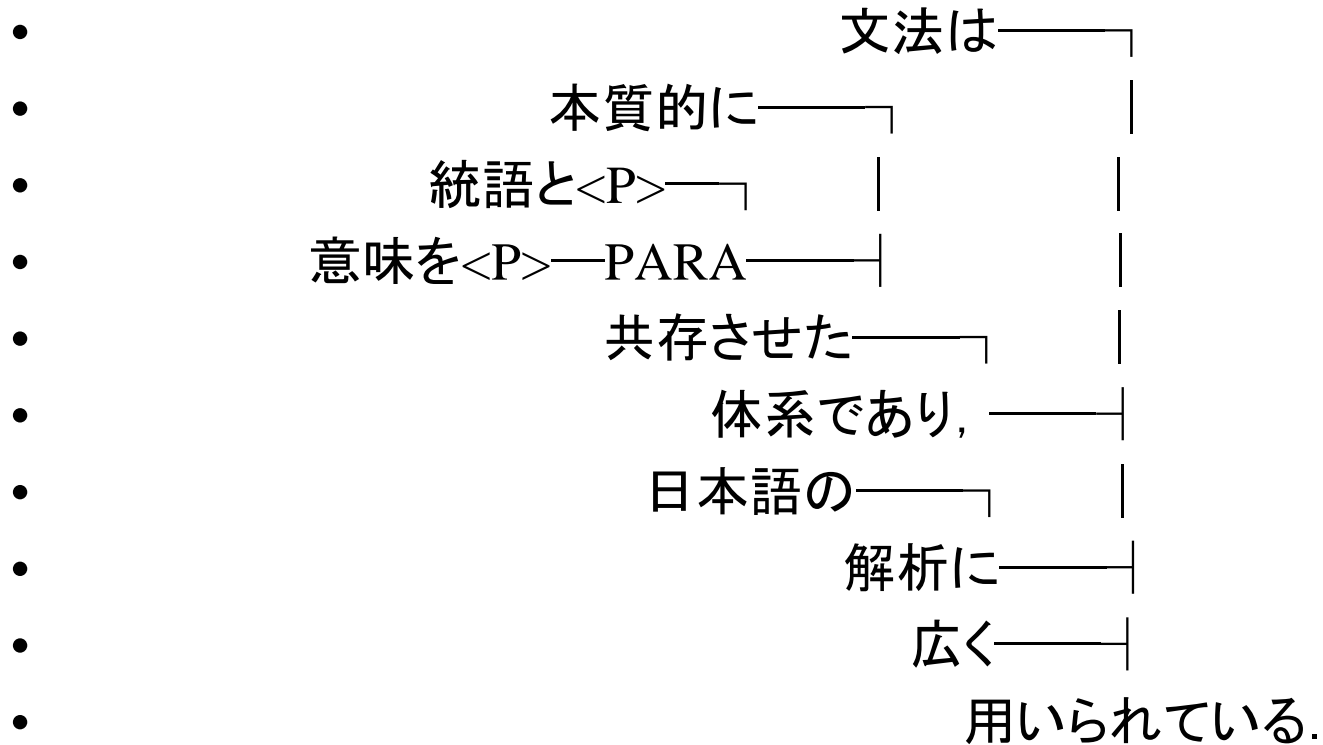
5. 文中に並列構造の可能性を示す表現があれば、その前後で類似する文節列を検出し、それを並列構造の範囲にする。
 1. 例:(((りんご(名詞))と(助詞(並列)))
((バナナ(名詞)) 並列構造)
が(格助詞))
6. 係り受け可能性の規則に沿って、文全体の係り受け構造の候補をつくり出す
 1. 並列構造は、主辞の文法的性質をもつまとまりとする。
7. つくり出された各候補を評価し、もっとも優先的なものを解として出力します。
 1. 基本的な評価基準は各係り受けの距離 と、各用言の表層格の充足度の総和

KNPの規則例

- (((係:デ格 読点) (係:カラ格 読点) (係:マデ格 読点))
 - ([((用言:強)) D])
 - ((レベル:C))
 - 2)
- ◆ **意味:** (係:デ格 読点)、(係:カラ格 読点)などのfeatureを持つ文節が(用言:強)というfeatureを持つ文節にD(通常の係り受け関係)で係り、(レベル:C)というfeatureを持つ文節があればそれ以上遠くの文節に係る可能性は考慮せず、最優先されるのは係り得る2番目に近い文節である

例

- 「文法は本質的に構文と意味を共存させた体系であり、日本語の解析に広く用いられている。」



語彙主導の単一化文法

- ◆ 英語などでは主語と動詞の性数一致なども要求される。
 - ◆ He stops → ○, He stop → ×
- ◆ これを全部書き換え規則で実現しようとする、膨大な数の規則になってしまう。
- ◆ 80年代から、書き換え規則の数を減らして、個別の単語に文法的性質(これを素性という)を与える方向に進んだ。
- ◆ 書き換え規則の適用は、単語の持つ素性が一致して初めて実行される。例えば、he = 名詞句(単数) と stops(単数)なら一致し he が stops の主語になることが分かるが、が、stop(複数)なら一致しないので、he は stop の主語ならない、という処理ができる。
- ◆ 一致の処理を単一化と呼び、このような文法を単一化文法と言う。

書き換え規則の爆発

- 簡単な規則からはじめよう。「太郎が走る」を解析するには
- 動詞句 → 後置詞句 動詞
- 「太郎がワインを飲む」を解析するには、自動詞と他動詞と分けて
 - 動詞句 → ガ後置詞句 自動詞
 - 動詞句 → ガ後置詞句 ヲ後置詞句 他動詞
- 「太郎が花子にワインを送る」という2つ目的語をとる他動詞の場合は
 - 動詞句 → ガ後置詞句 自動詞
 - 動詞句 → ガ後置詞句 ヲ後置詞句 他動詞
 - 動詞句 → ガ後置詞句 二後置詞句 ヲ後置詞句 他動詞

書き換え規則の爆発

- 「太郎がワインを花子に送る」となると
 - 動詞句→ガ後置詞句 自動詞
 - 動詞句→ガ後置詞句 ヲ後置詞句 他動詞
 - 動詞句→ガ後置詞句 ニ後置詞句 ヲ後置詞句 他動詞
 - 動詞句→ガ後置詞句 ヲ後置詞句 ニ後置詞句 他動詞
- しかし、「太郎はφ花子にワインを送る」
- φは省略されたガ格でゼロ代名詞。「太郎は」は主題
 - 動詞句→ ハ後置詞句 ヲ後置詞句 ニ後置詞句 他動詞
- も追加。しかし、ハ後置詞句はいろんなところに現れうる。
全部、書き換え規則で書いては. . . .

そこで

- 言語現象の複雑な部分を書き換え規則数を増やすことではなく、個々の単語の意味に埋め込む方向が模索された。
 - 一般化句構造文法 (Generalized Phrase Structure Grammar: GPSG)、
 - 語彙機能文法 (Lexical Functional Grammar: LFG)、
 - 主辞駆動文法 (Head driven Phrase Structure Grammar: HPSG)
- という一連の文法体系が80年代から提案され、計算機での構文解析の基礎理論になった

単一化文法のひとつである主辞駆動文法

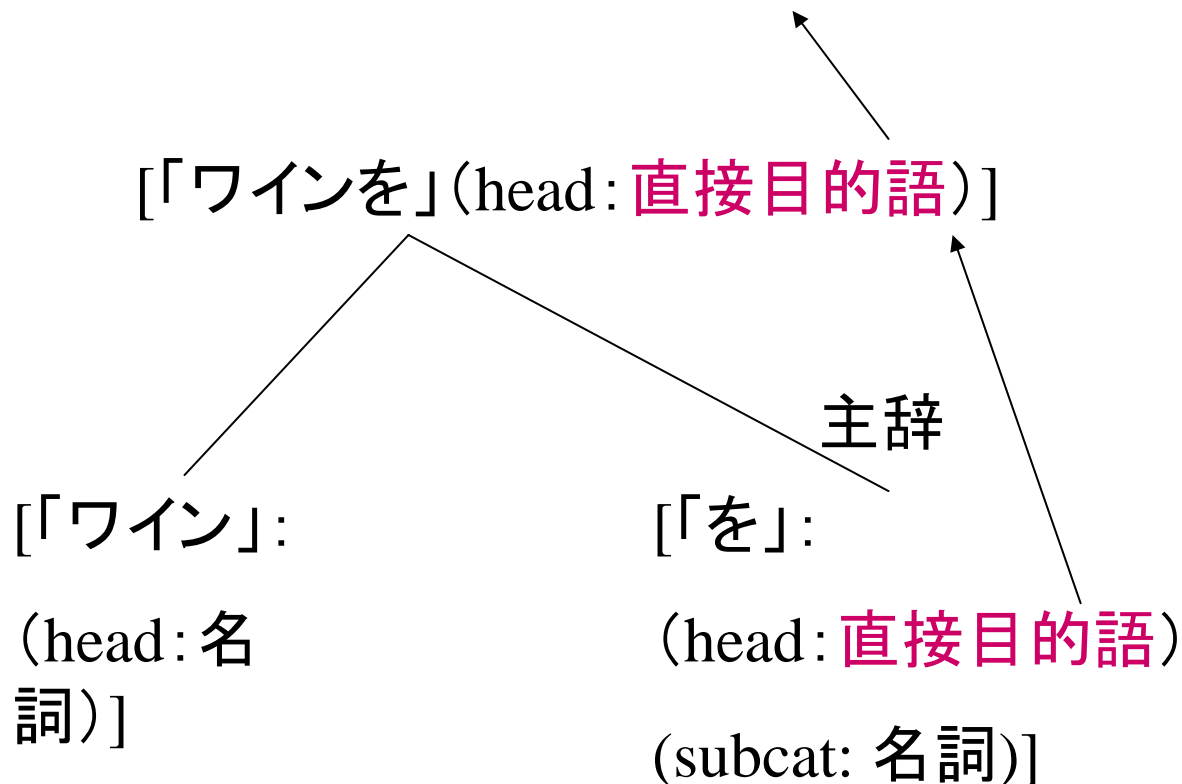
- ◆ 主辞(head)とは
- ◆ 「ワインを」で「ワイン」と「を」のどちらが文法的に大切か？
 - ◆ 「ワインを飲む」OK
 - ◆ 「ワインに飲む」×
 - ◆ 「ワインが飲む」×
- ◆ 後ろにくる動詞との文法的関係は「ワイン」ではなく「を」が担っている。つまり、後置詞句「ワインを」の文法的な主体は「を」である。このような文法的主体を**主辞**という。
- ◆ 主辞にいろいろな文法的性質を記述し、主辞がその文法情報を構文木の隣接ノードに伝える形の文法体系

素性(そせい)と下位範疇化

- ◆ 語の文法的性質はいくつかの要素によって表される
- ◆ 要素を素性(feature) という。
- ◆ 例えば、人称代名詞「彼」は
 - ◆ 「代名詞」
 - ◆ 「照応(他の要素を参照する)」
- ◆ という素性を持つが
 - ◆ 「再帰的」という素性は持たない
 - ◆ 「彼自身(himself)」なら再帰的という素性を持つ
- ◆ ある語彙がどのような補語をとるかを**下位範疇化**(subcategorization)という。つまり、その語彙の文法的役割をより細かく(これが「下位」の意味)分類する情報のこと。
- ◆ 例えば、「飲む」という動詞は
 - ◆ ガ格主語(格助詞「が」をもつ後置詞句)と
 - ◆ ヲ格目的語(格助詞「を」をもつ後置詞句)
- ◆ を下位範疇化素性として持つ

主辞素性原理

- ◆ 任意の句の主辞の持つ文法的性質すなわち主辞素性の値はその句の主辞の主辞素性の値に等しい.
- ◆ Headとは主辞素性のこと



下位範疇化による語彙の記述例

□ 自動詞 :

[head : 動詞 , subcat {後置詞句(が)}]

□ 他動詞1 :

[head : 動詞 ,
subcat {後置詞句(が), 後置詞句(を)}]

□ 他動詞2 :

[head : 動詞 ,
subcat {後置詞句(が), 後置詞句(を),
後置詞句(に)}]

□ 後置詞 :

[head : 後置詞句(格助詞), subcat {名詞句}]

下位範疇化素性が記載された単語が別の単語や句とつながって上位の句や文を形成していく過程で下位範疇化素性がどのように変わっていくかを記述したのが下位範疇化の原理

■下位範疇化の原理

- 句全体の下位範疇化素性の値は主辞の下位範疇化素性値から補語を取り除いたものに等しい。
- この原理を他動詞にその補語として目的語である後置詞句をつなげて文を組み立てる例で見よう

下位範疇化原理の適用例

- 名詞と格助詞から後置詞句を組み立てる過程（簡単のため、格助詞は下位範疇化素性として名詞をとることにしておく。）
 1. を : [head : 後置詞句(を), subcat { 名詞 }]
 2. ワインを : [head : 後置詞句(を), subcat { }]
- 次に後置詞句と動詞を組み合わせる過程を示す。
 3. 送る : [head : 動詞, subcat { 後置詞句(が), 後置詞句(を), 後置詞句(に) }]
 4. ワインを送る : [head : 動詞, subcat { 後置詞句(が), 後置詞句(に) }]
 5. 花子にワインを送る : [head : 動詞, subcat { 後置詞句(が) }]
 6. 太郎が花子にワインを送る : [head : 動詞, subcat { }]

単語の意味と上記の原理にほとんどの作業が任されたため、当初の目的通り書き換え規則の数は極少数になった。

例えば、名詞、格助詞、動詞から動詞句を組み立てる規則は次の三つだけである。

◆後置詞句 → 名詞 後置詞

◆動詞句 → 後置詞句 動詞

◆動詞句 → 後置詞句 動詞句

◆この書き換え規則で上の例である動詞句の一部「ワインを送る」が構文解析されてできる構文解析木を図示

動詞句 : [head: 動詞, subcat { 後置詞 (が), , 後置詞句
(に) }]
後置詞句 (を)

[head: 後置詞句 (を), subcat { }]

名詞: ワイン [subcat { 名詞 }]

後置詞: [head: 後置詞句 (を), subcat: { 名詞 }]

ワイン

を

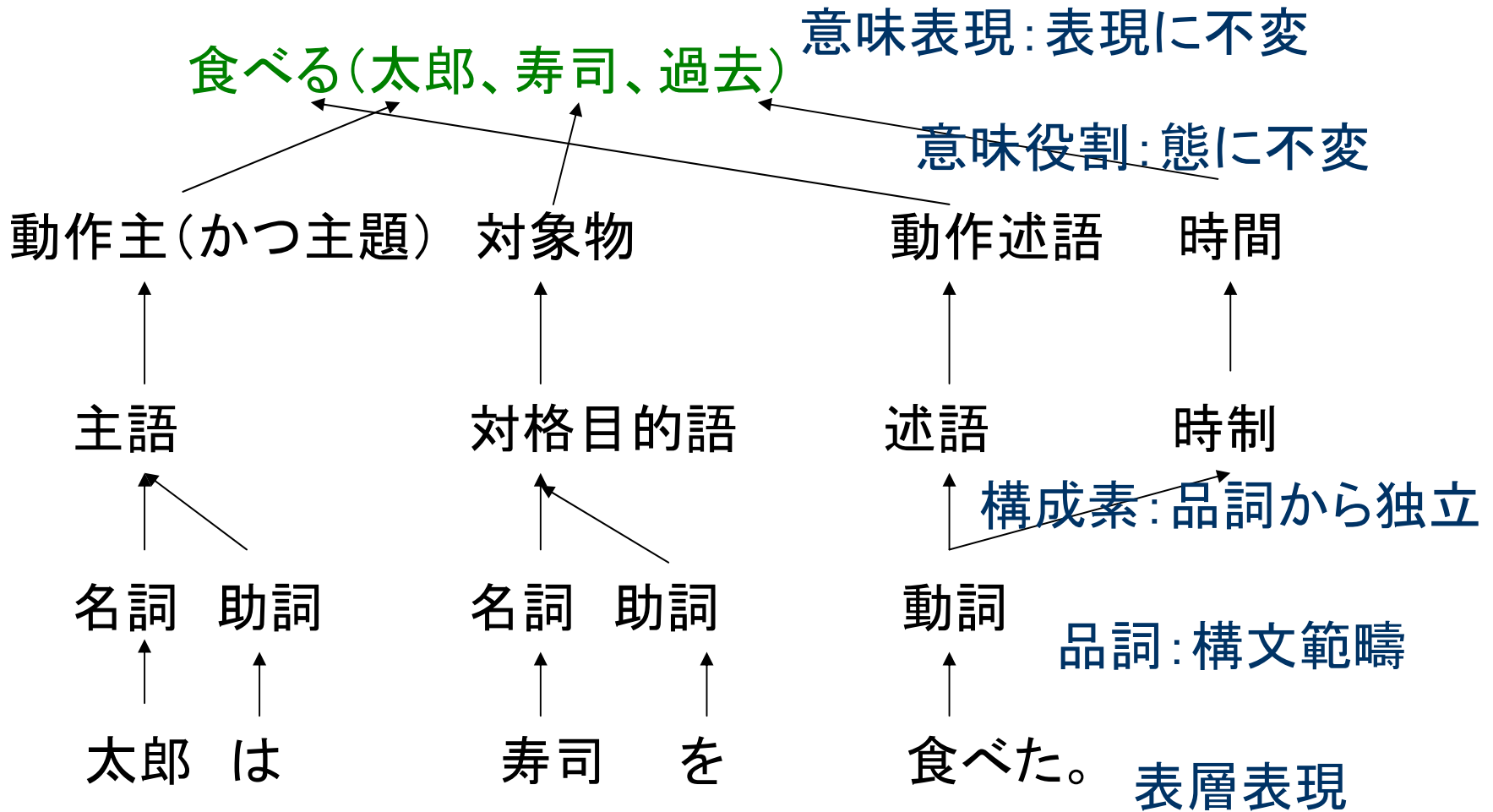
送る

[head: 動詞, subcat { 後置詞 (が)、
後置詞句 (を), 後置詞句 (に) }]

意味の表現

- 動詞、名詞 といった品詞(構文範疇)
- 主語、述語 といった文法役割(構成素)
- 動作主、対象物 といった意味役割
 - 意味役割は、言語表現に対して現実世界のモノやコトに対応するもの。
 - 微妙な役割として、主題、被害者(被害受身): 語用論的役割
- 貸す(太郎、花子、本) というような意味の命題表現
- これらの間の関係を付ける作業こそ広い意味で文法の仕事
- 計算機における言語のモデル化でも重要

言語表現から意味へのつながり

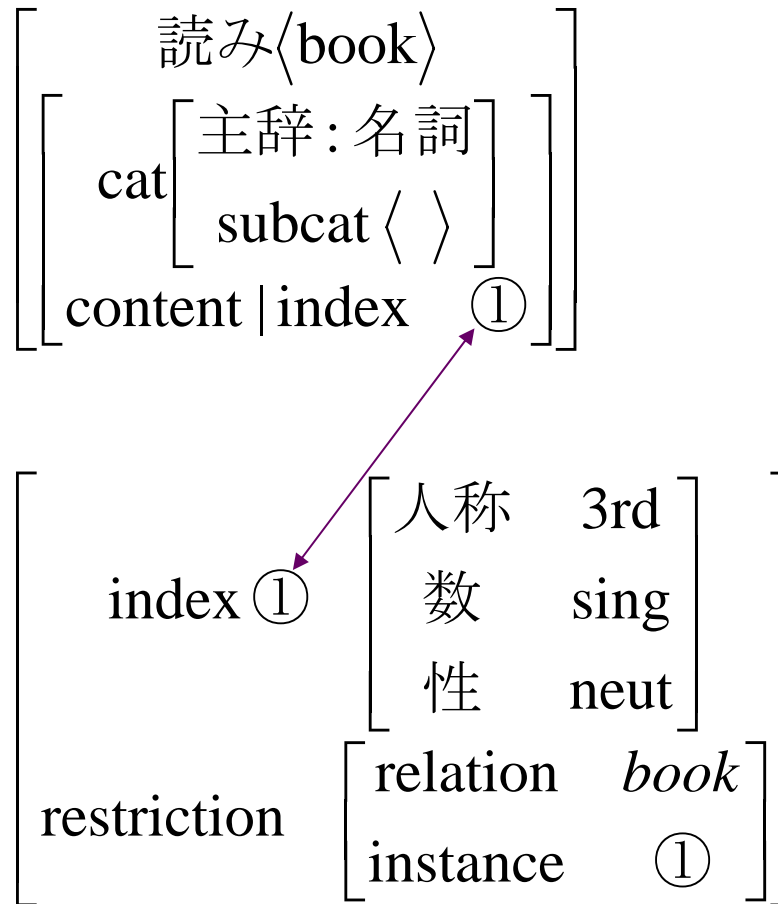


各レベルの関係

- 下位範疇化では、品詞をsubcatで要求していた。
- 文法役割(構成素:これは格(主格、対格、与格など))がどのような意味役割に結びつくかが言語学の大きなテーマ
- θ 役割(チョムスキー理論)は、意味役割といえるが、微妙に文法役割と絡む
 - 主題(テーマ)は θ 役割だが、これは意味役割というよりは文法的概念。
 - θ 役割と文法的格との対応付けの精緻な理論としての言語学

HPSGでの文法、意味の統一的表記法

- 名詞の素性構造表現の例



HPSGでの文法、意味の統一的表記法

動詞 gives の例

		読み⟨gives⟩
		主辞: 動詞[finite]
cat		subcat ⟨名詞句[主格] _{①[3rd sing]} , 名詞句[与格] _② , 名詞句[対格] _③ ⟩
content	relation	<i>give</i>
	giver	①
	given	②
	gift	③
context	speaker	①

モノの意味

- 述語の項となるモノにはいろいろな意味役割がある。
 - 現実世界にけるモノとしての意味が述語の意味と関連して、述語の項としての役割になる。(後述)
- まずはモノそのものの意味
- モノ(そしてコト)の意味が織り成す階層構造を記述するのがシソーラス
- 実際の意味は詳しく記述しようとするれば、単語そのものまで分解されてしまう。そこでできるだけ独立した意味の成分(=意味素性)に分解する。

モノの意味を構成する意味素性

IPAの基本名詞辞書による

- 動物の領域 ANI
 - HUM(人間)、AML(人以外の動物)
- 具体物の領域 CON
 - AUT(自立的 ex コンピュータ), EDI(食物)、LIQ(液体), PAS(粘性), SOL(固体)
 - 米飯: EDI&PAS&SOL、ビール: EDI&LIQ
- 場所の領域 SPA
 - LOCUS(移動の始点、終点となる場所), INT(内部があるもの ex 部屋), ORG(組織), NET(ネットワーク ex 交通網)

意味素性 つづき

- 出来事および動作、作用の領域：PRC
 - ACT(動作、行為)
 - EVE(イベント、出来事)
 - APO(予定に従った行動：ex 銀行が9時から始まる)
 - RES(結果 ex 災害)
 - PRO(結果、制作物 ex パンを焼く)
 - PHE(自然現象の結果できるもの ex 氷が張る)
 - NAT(自然物、現象 ex 台風、太陽)
 - PLA(植物)
 - GAS(気体 ex 霧、息)
 - ELM(五感では捉えられない性質 ex たんぱく質、神経)
 - POT(身体部位 potency ex 足、肩、肺、胃腸)

意味素性 つづき

- 抽象性の領域: ABS
 - Price (収入、価格)
 - Measure (身長、体重)
 - Information (情報、身長、小説、音楽、批評、住所)
 - Quantity (重量、面積)
 - Social bonds (格差、関係)
 - Grade (身分、評価、規模)
 - Form (評価される属性 ex 味、形)
 - Attribute (程度で計るもの ex 非常識、進歩、塩)
 - Reciprocity (相性)
 - Personality (意地、性格)
 - Mind (勘、神経)
 - Manner (能力、性向など ex 料理、詰め、発表、運転、色使い、人使い)

意味素性 つづき

- 抽象性の領域: ABS
 - Method(方法、やり方)
 - Objective-value(値 ex 赤、四角)
 - Sensational-value(甘い、辛い)
 - Evaluation(評価 ex 台所が苦しい、財政、舌)
 - Currency(価格 ex 100ドル、1000円)
 - Duration(期間 ex 3年)
 - Distance(距離 ex 3km)
 - Item(数を表す、ex 3人、1個)
 - Ratio(割合, ex 30%)
 - Quantity(量 ex 30kg)
 - State(状態 ex 安定、幸福、不幸、静か、可能、頑固)

意味素性 つづき

- 抽象性の領域: ABS
 - Role(役職名)
 - Relational-term(親族、交友関係)
 - Direction(東西南北左右上下前後)
 - Phase(時間的、位置的順序)
 - Reference-point(基準点からの相対 ex 逆、以上)
 - Norm(規則、法則、法律、公式)
 - Subfield(学問、芸術、スポーツなどの分野)
 - Inclination(心理的傾向 ex 興味、馴染み)
 - Appearance(外見 ex 印象、態度、形跡)
 - Unit(単位)
 - Time-point(時点)
 - Time(出来事の順序関係、抽象的時間 ex 将来)

意味素性 つづき

- 抽象性の領域: ABS
 - Ordinal(順序、順番)
 - Name
 - Entity
 - Congregation(集合物 ex 群れ、世間、有志)
 - Kind(種類 ex 人類)
 - Abstract(その他の抽象概念)

動詞の項の意味役割 (人間的なもの)

➤ 動作主(agent)

- 動作を行ったモノ。意図を持つことができるモノでなければならない。よって、使役者ともなる。

➤ 行為者(actor)

- Actionを行ったモノ。ただし、意図的でない行為でもよい。使役者にはなれないとされる。

➤ 非動者(patient)

- 動作を受けたモノ

➤ 経験者(experiencer)

- ある状態になった(=経験した)モノ

例

- 太郎は走った。
 - 走る (actor=太郎, time<now)
- 花子はりんごを食べた。
 - 食べる (agent=花子, patient=りんご, time<now)
- 次郎は驚いた。
 - 驚く (experiencer=次郎, time<now)
- 太郎は次郎を驚かせた。
 - cause (agent=太郎, patient=次郎、
驚く (experiencer=次郎), time<now)

動詞の項の意味役割

➤ 対象(object)あるいはモノ(thing)

➤ 動作の対象物。

➤ 場所(place)

➤ 動作の行われた場所。

➤ 結果(resultant)

➤ 状態(state)

➤ stateは中立的な状態だが、resultantは動詞の記述する動作の結果として生まれるstate。ex. bear→alive。そうではないstateとしてはsleepy,hungry。

動詞の項の意味役割の細分化 (以下のいくつかの意味素から構成)

- 対象(object)あるいはモノ(thing)
 - 動作の対象物。 physical object vs abstract object の対立あり
- 場所(location)
 - 動作の行われた場所。
- 起点(source)
 - 動作を開始した場所。
- 終点(goal)
 - 動作が終了した場所。
- 方向(direction)
- 経路(path)
- 道具(instrument)
 - Taro goes to school from his house by bike on route101.
 - go(actor=Taro, source=his house, goal=school, path=route101, instrument=bike, time=now)

動詞の意味素分解

- 基本的な意味素に分解して考える。
 - affect(影響) affect(actor, patient)
 - effect(結果) effect(actor, resultant)
 - act(行為) act(actor, X), X=patient, object, ...
 - experience(経験) experience(experiencer, state)
 - order(命令) order(agent, action(actor)),
 - より細分化したレベルでは
- be, move, cause, alive, die, see, hear, have, eat, sleep, sell, buy,
- 意図性、などメタレベルな意味素 : volitional
- 例:
 - 殺す : kill(actor, patient) = cause_{volitional}(actor(not(alive(patient))))
 - 産む : kill(actor, patient) = cause_{volitional}(actor(alive(patient)))
- 例: 太郎に学校へ行くけと命ずる
 - order(agent=X, act:行く(actor=太郎, goal=学校))

コトから陳述へ その1

- コトの意味とはおよそ述語、具体的には動詞の意味に項の意味を組み合わせる出来上がる。
- 話し手の主観を排したコトを命題 proposition と呼ぶ。
 - コト: 命題の例: 太郎が次郎を殴る
 - beat(agent=太郎, patient=次郎)
- 話し手の命題に対する態度を陳述という。
Speakers attitude towards the proposition あるいは modality
 - 例えばコトの生起した時刻と発話時刻の関係は(過去、現在、未来)という時制で表される
 - 太郎が次郎を殴った
 - beat(agent=太郎, patient=次郎, time < ST)
STとはspeech time)

コトから陳述へ その2

- 話し手の命題に対する態度を陳述という。
Speakers attitude towards the proposition あるいは modality
- 命題に対する態度としては、推測(だろう:主観的、らしい:伝聞的)、伝聞(そうだ)
 - 太郎が次郎を殴ったそうだ。
 - 伝聞(beat(agent=太郎, patient=次郎, time< ST)
news-source=X)
- 文の意味はそれが話された文脈においては (陳述(命題)) という入れ子構造を持つ。

陳述における語用論的役割

- 主題 (topic, theme)
 - 文でもっとも問題にしているモノ、コト
- 話者 (speaker)
 - 伝聞している人というのもある。“彼によればもう終わったらしい。”
- 聞き手 (hearer)
 - 終助詞「よ」「ね」などによってimplicitに導入される。

聞き手への態度へ

- 主題: 命題に対する話し手の評価であるが、聞き手への伝えたいコトの順位付けとも考えられる。
 - 太郎は次郎を殴ったそうだと。
 - (topic=太郎, 伝聞(beat(agent=topic, patient=次郎, time< ST)))
- 聞き手への態度: (命題+モダリティ)を聞き手にどう伝えるか。質問「か」、念押し「よ」、確認「ね」
 - 太郎は次郎を殴ったそうだね。→聞き手に命題内容を確認。命題内容は聞き手から聞いたことかもしれない。
 - ね(話し手、聞き手、伝聞(beat(agent=太郎, patient=次郎, time< ST) news-source=?聞き手))
 - 太郎は次郎を殴ったそうだよ。→聞き手に命題内容を確実に伝える(念押し)
 - よ(話し手、聞き手、伝聞(beat(agent=太郎, patient=次郎, time< ST) news-source=X)) Xは 聞き手ではない。

時間 (Reihenbach)

- 発話時刻:ST speech time
- 事象の生起時刻:ET event time
- 参照時刻、(話し手の視点): RT reference time
 - 過去完了なら $ET < RT < ST = \text{now}$
- Tense: ST, RT, ETの前後関係
- Aspect: 事象の継続の時間軸上で区間として指定する。英語のing, -ed の意味に現れる
 - 継続、繰り返し、終了、影響が残るなどなど。

日本語の tense, aspect

- **1次aspect** : Tense: 未然:「する」vs 已然「した」
 - 「する」が未来を表す。「ぼくは勉強する」はこのままで “I will study.” 「する」が意志も表せる。
- Aspect: 「ている」は時間的aspectで完了と継続の曖昧さ。
 - cf. (する vs した) vs (している vs していた)
- **2次aspect**: ところが、「している」に見られる「動詞 + て + *」という形式は 動作主の意図、対象物の性質など時間的 aspect 以外の情報を担う。
 - 日本語では、この形が豊富である。
- **3次aspect** さらにある種の動詞は別の動詞に後接して複雑な意味を形勢する。
 - 「しはじめる」など
- 態度やモダリティを表す助動詞につながる。

2次aspect

➤ 「ている」

- 継続動作の動詞＋ている： 進行中
 - 「読んでいる」「食べている」
- 瞬間動作の動詞＋ている： 結果の残存
 - 「死んでいる」「決まっている」「終わっている」
 - 動詞が瞬間、継続の両義なら「ている」もまた両義
- 状態を表す動詞＋ている： 状態にあることの強調
 - 「そびえている」「ばかっている」

➤ 「てしまう」 完了の強調

➤ 「ていく」「てくる」

- 動作主から見た(空間的、時間的)方向性を示す。
- 「見に行く」「買ってくる」

2次aspect

- ▶ 「**てある**」 過去の意図的行為の結果
 - ▶ X 病気になってある
 - ▶ ただし、「CDが捨ててある」のように行為者が不明、意図も不明の場合に使える。
- ▶ 「**ておく**」「**てみる**」「**てみせる**」 動作主の意図による行為
 - ▶ 結果よりは行為そのものに焦点
 - ▶ 「てみせる」受け手を強く意識する
 - ▶ X 痒くなってみせる。 Cf, ○痒が**って**みせる。
- ▶ 「**てやる/てあげる**」「**てもらう/ていただく**」「**てくれる/てくださる**」
 - ▶ 行為者と恩恵を受ける者との関係を表す。

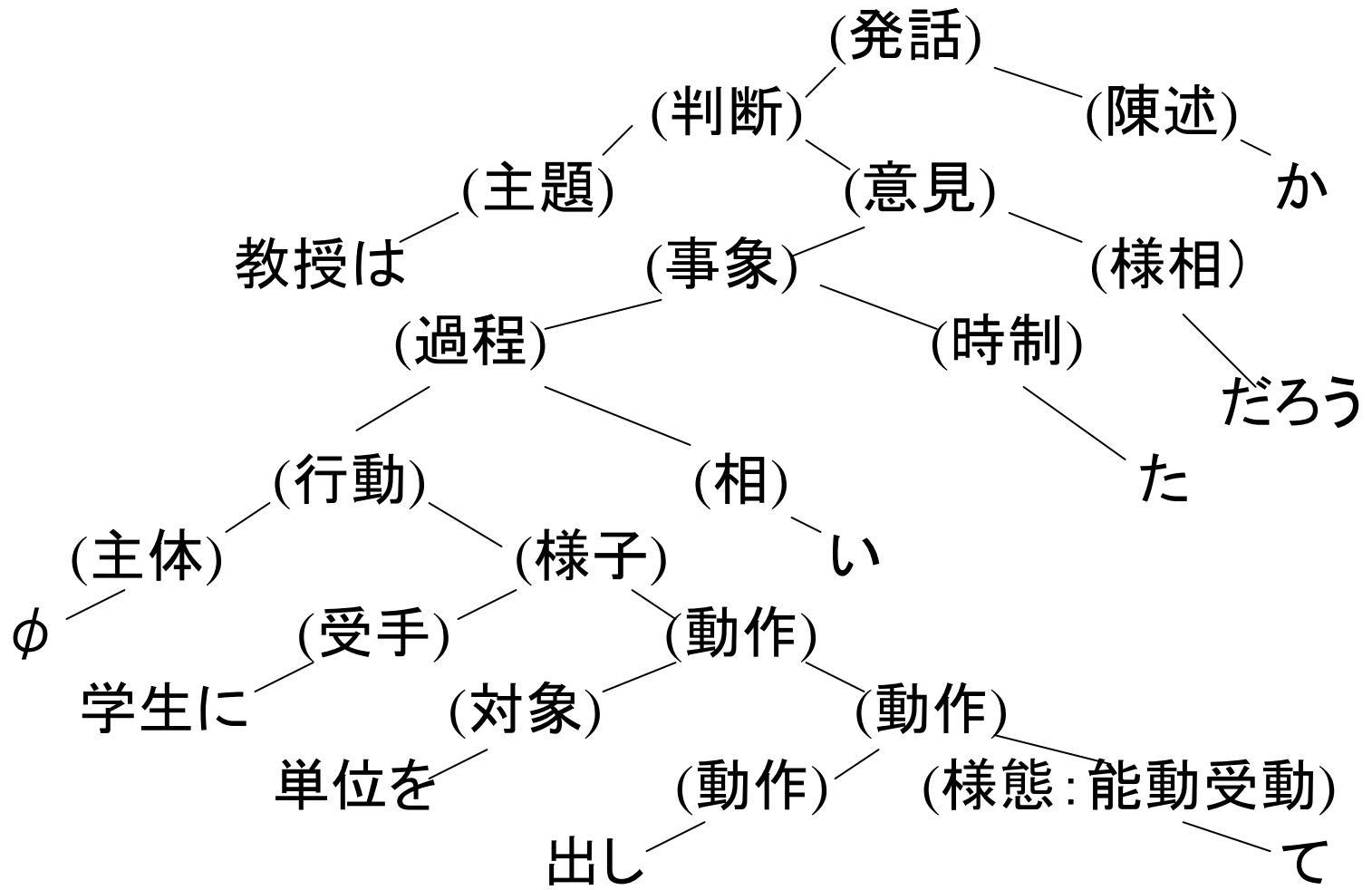
3次aspect

- 単独で動詞として使われるいくつかの動詞(補助動詞)がある種の動詞(本動詞)に後接して複合動詞となる
- 「はじめる」「こむ」「だす」「あう」「つづける」「かける」「あげる」「きる」「つける」「つく」
- さらにその後ろに2次+1次のaspectを伴う。
- 意味的には本動詞と補助動詞の意味を合成する。
 - 多くは(補助動詞の意味(本動詞の意味))という意味構造

日本語の発話の構造

- 詞と辞 時枝誠記
 - 詞＝客観的、辞＝主観
 - ((詞)辞) あるいは 句→詞 辞, 詞→詞 辞
 - 入れ子構造
- Proposition and Modality
- 命題 (proposition) の構造
 - 主語と述語
 - 動作主と動作
 - 経験者と感覚

日本語の発話の構造 (郡司隆男)



教授はφ学生に単位を出していただくだろうか