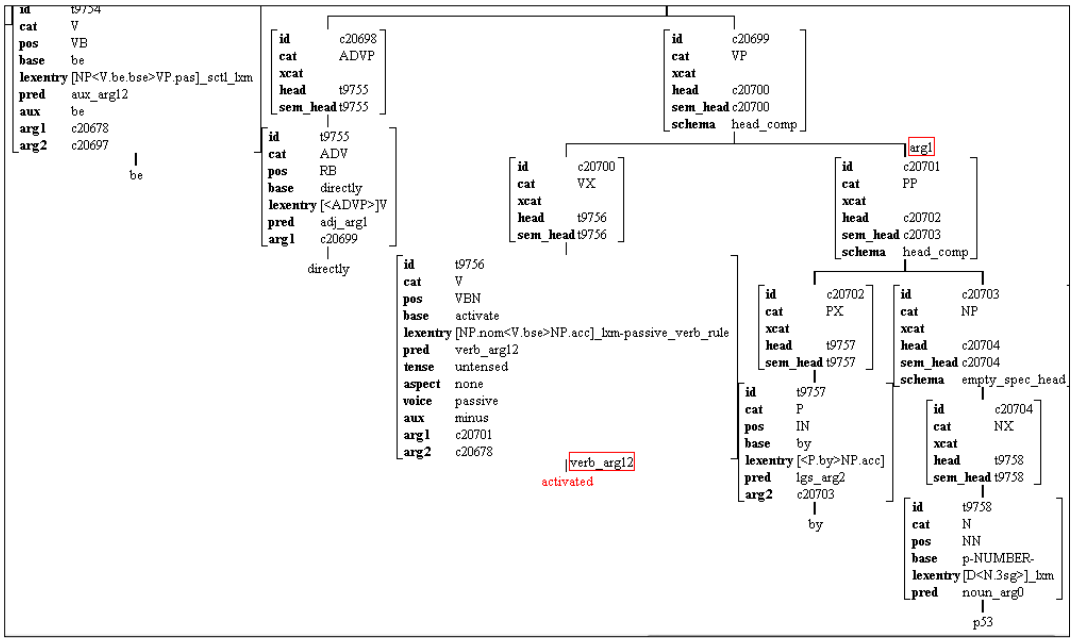
自然言語処理における構文解析と言語理論の関係

宮尾祐介
国立情報学研究所

研究紹介

- 言語理論に基づく構文解析
- 英語HPSGパーザ Enju
  - HPSG理論に基づき統語構造（構文木）と意味構造（述語項構造）を計算
- 中国語、日本語も



Enju の出力

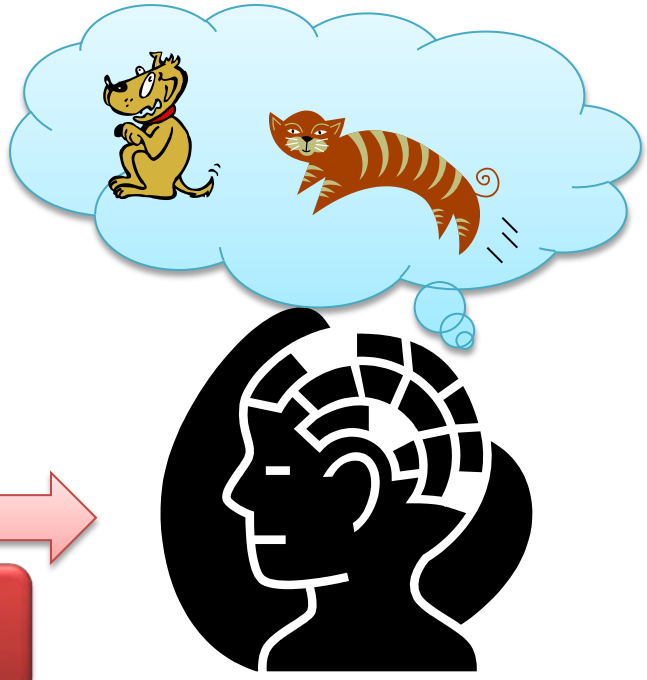
- 述語項構造

- 述語論理式
  - Stately, plump Buck Mulligan came from the stairhead, bearing a bowl of lather on which a mirror and a razor lay across. A yellow dressinggown ungirdled was sustained gently behind him on the mild morning air.
  - $\exists v\, x_1\, x_2\, x_3$ (plump_buck_mulligan(z) ∧ lather(x) ∧bowl+of(x_2, x) ∧ mirror(v) ∧lay_across+on(v, x_2) ∧ razor(x_1) ∧ lay_across+on(x_1, x_2) ∧ $\text{bear}_{\text{prog}}$($z$, x_2) ∧ stairhead(y) ∧ $\text{stately\_come}_{\text{past}}$+from($z$, y) ∧ yellow_dressinggown_ungirdled (x_3) ∧ mild_morning_air(u) ∧∃ x_4 (x_4 = z ∧$\text{sustain}_{\text{past,passive}}$_gently+behind+on ($x_3$, x_4, u)))

自然言語の構文解析

- テキストを入力として、その意味表現を計算する

出力:意味表現



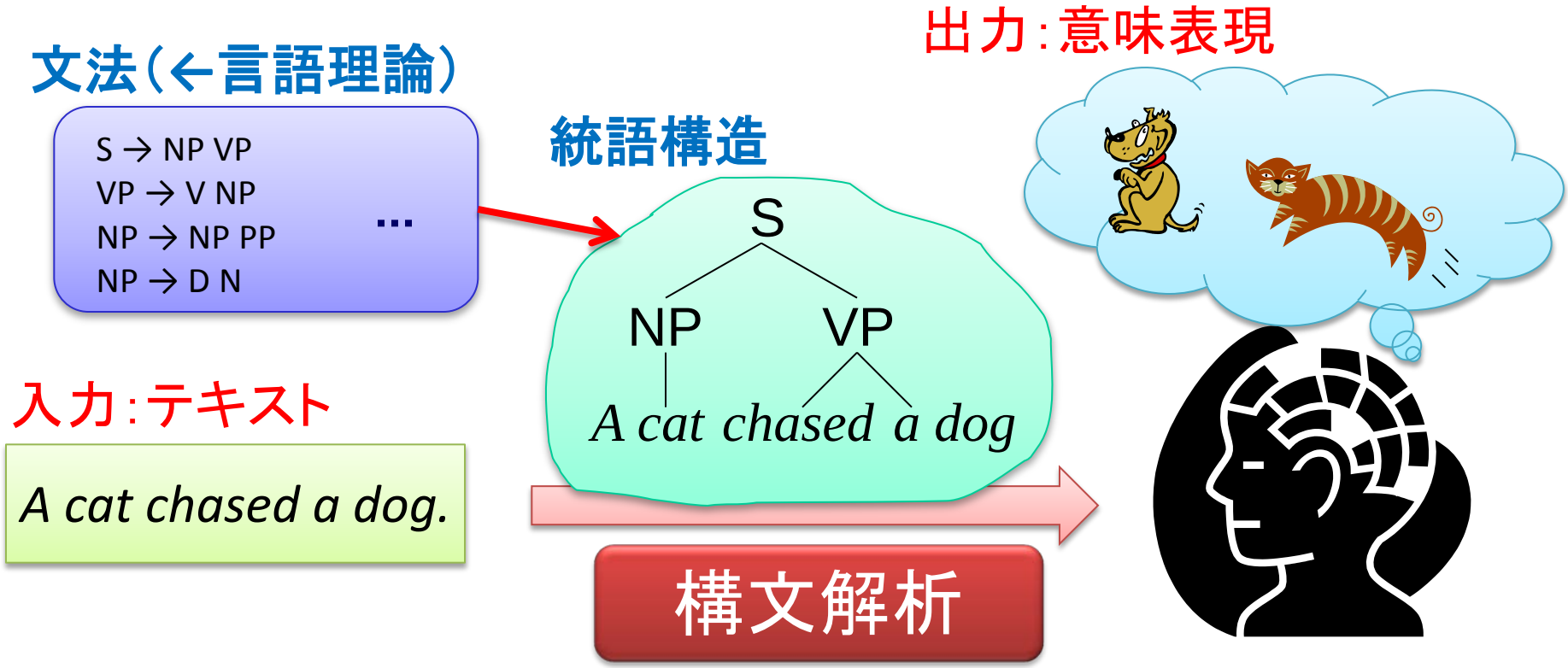
入力:テキスト

A cat chased a dog.

構文解析

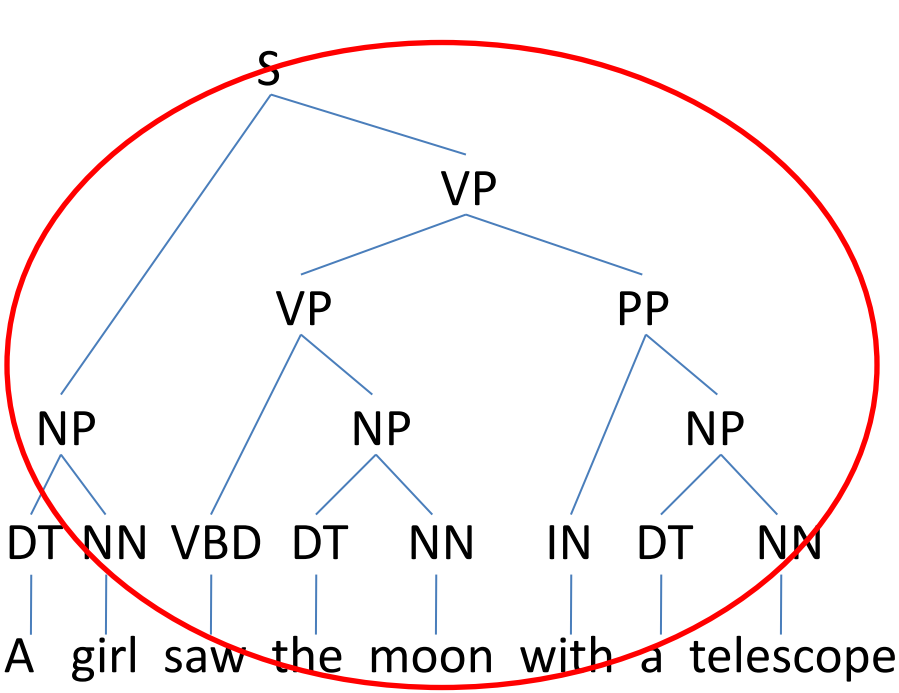
アプローチ

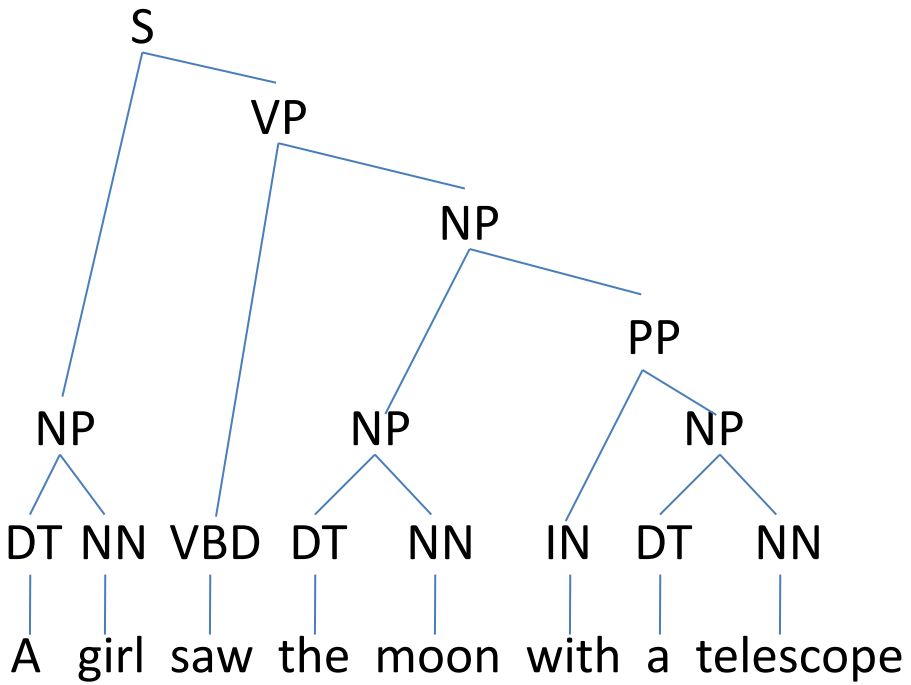
- **統語構造**：文と意味表現の橋渡し
- **文法**：構文構造を導出する規則
  → 文法を実装すれば、構文解析ができる！



構文解析と文法

- 文法を実装すれば、構文解析ができる → 間違い
- 文法は文法的に適格な構造を規定する
- 文法的に適格な構造の集合 ≫ 人間の解釈
- 構文解析では、人間の解釈に相当する構造が欲しい





構文解析と曖昧性解消

- 曖昧性解消がない構文解析は意味がない
  - 構文解析 ＝ 探索問題
  - 高い構文解析精度＝ よい曖昧性解消
- 自然言語処理における構文解析研究のほとんどは曖昧性解消(≒統計モデル)について
  - 逆に、文法の役割は忘れられている
- 今日は、構文解析において文法や言語理論が果たす役割について議論

範疇文法(CG)

- カテゴリを組み合わせることで統語構造を計算
- 基本カテゴリ: S, NP, N
- 複合カテゴリ
  - カテゴリを “/” や “\” でつなげたもの
  - X/Y は、カテゴリ Y が右側に来るとくっついて X になる
  - X\Y は、カテゴリ Y が左側に来るとくっついて X になる
- 例:
  - 自動詞: S\NP
  - 他動詞: S\NP/NP

NP	S\NP	→	S
John	*walked*		

規則

- 関数適用規則
  - X/Y　Y　⇒　X　　(>)
  - Y　X\Y　⇒　X　　(<)
- 例:
  - S/NP　NP　⇒　S
  - NP　S\NP　⇒　S

NP	S\NP	→	S
John	*walked*		

構文解析

- 各単語にカテゴリ（語彙カテゴリ）を割り当て

NP	S\NP/NP	NP
John	*loves*	*Mary*

構文解析

- 規則を適用し、カテゴリをくっつける

$$X/Y \quad Y \quad \Rightarrow \quad X \qquad (>)$$

X = S\NP, Y = NP

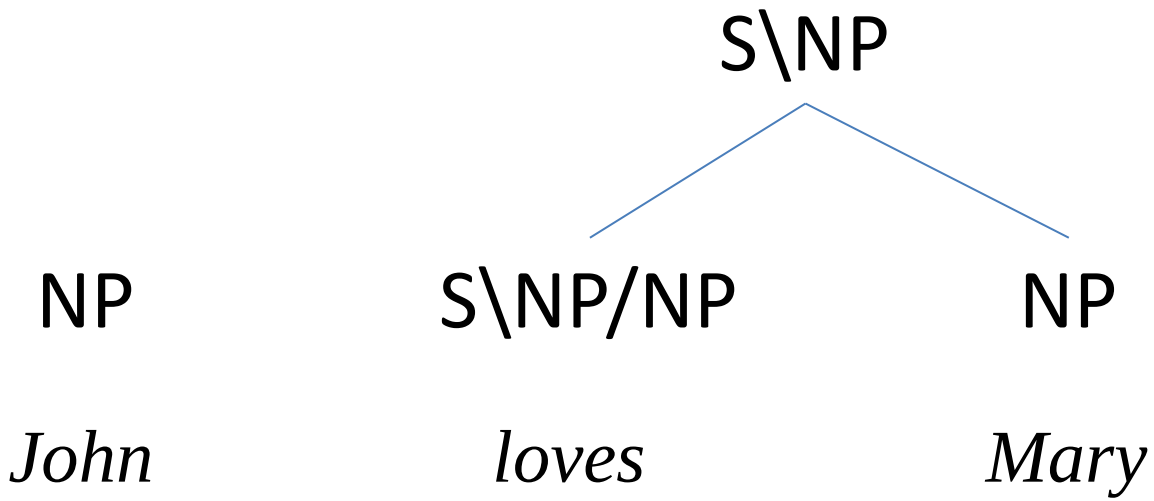
NP	S\NP/NP	NP
John	*loves*	*Mary*

構文解析

- 規則を適用し、カテゴリをくっつける

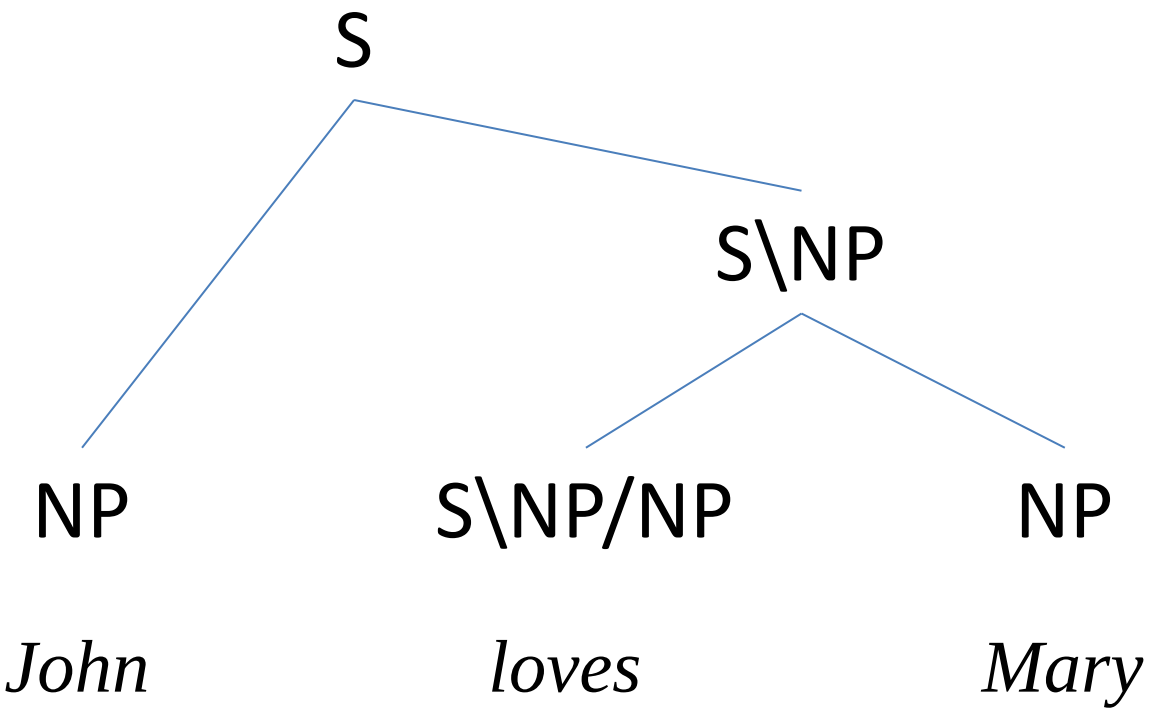
Y　X\Y　⇒　X　　(<)

X = S, Y = NP



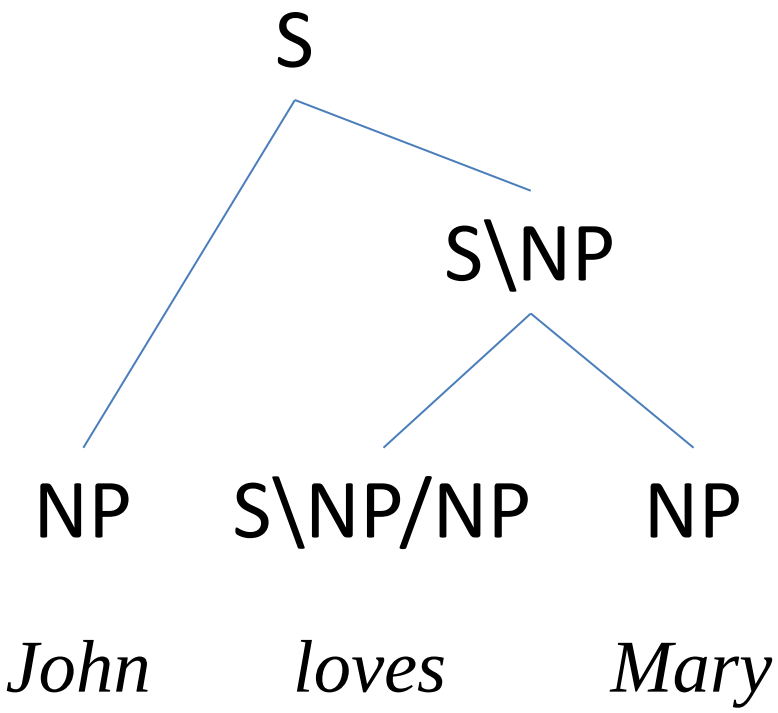
構文解析

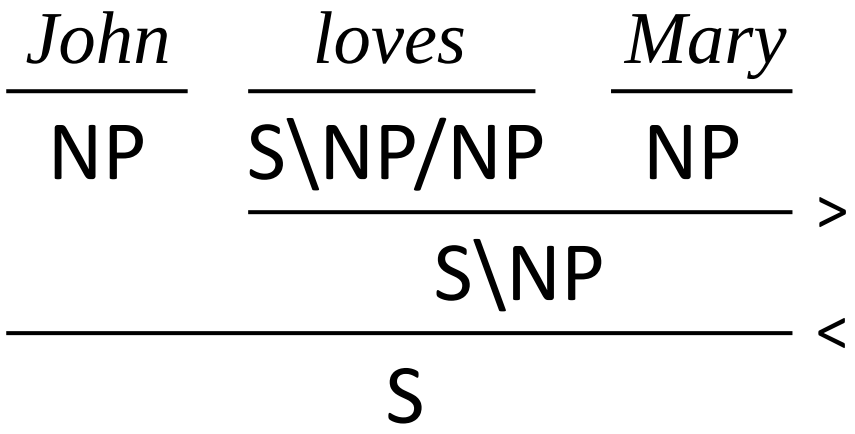
- 規則を適用し、カテゴリをくっつける



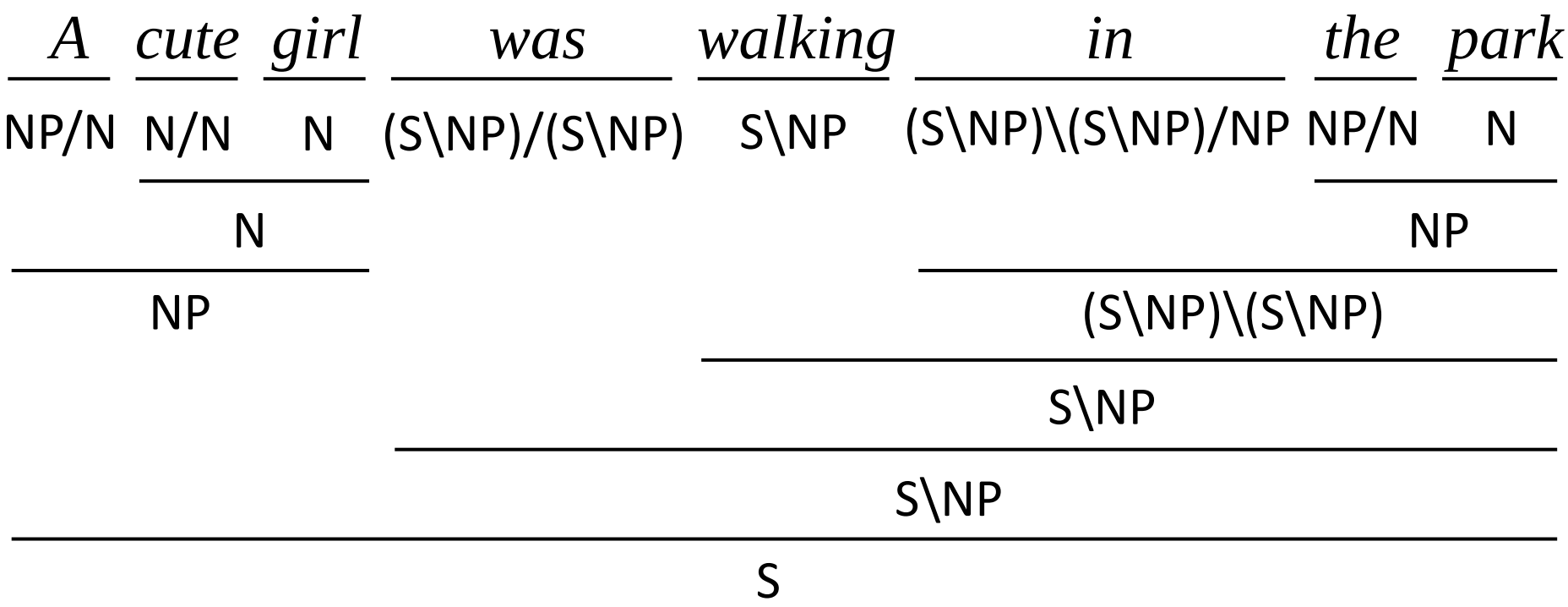
構文解析 = 証明

- 範疇文法の分野では、伝統的に構文解析を証明として表す





ちょっと難しい例


A cute girl was walking in the park
NP/N N/N N (S\NP)/(S\NP) S\NP (S\NP)\(S\NP)/NP NP/N N
N
NP
NP
(S\NP)\(S\NP)
S\NP
S\NP
S

意味構造の計算

- 各カテゴリには、意味構造のλ式が割り当てられる
- 統語構造に沿ってλ計算を行うと、意味構造が計算できる

John	*loves*	*Mary*	
NP: john	S\NP/NP: λx.λy.love(y, x)	NP: mary	
	S\NP: λy.love(y, mary)		>
S: love(john, mary)			<

組合せ範疇文法(CCG)

- 範疇文法に数個の規則を追加することで、
  自然言語の様々な構文がすっきり解析できる

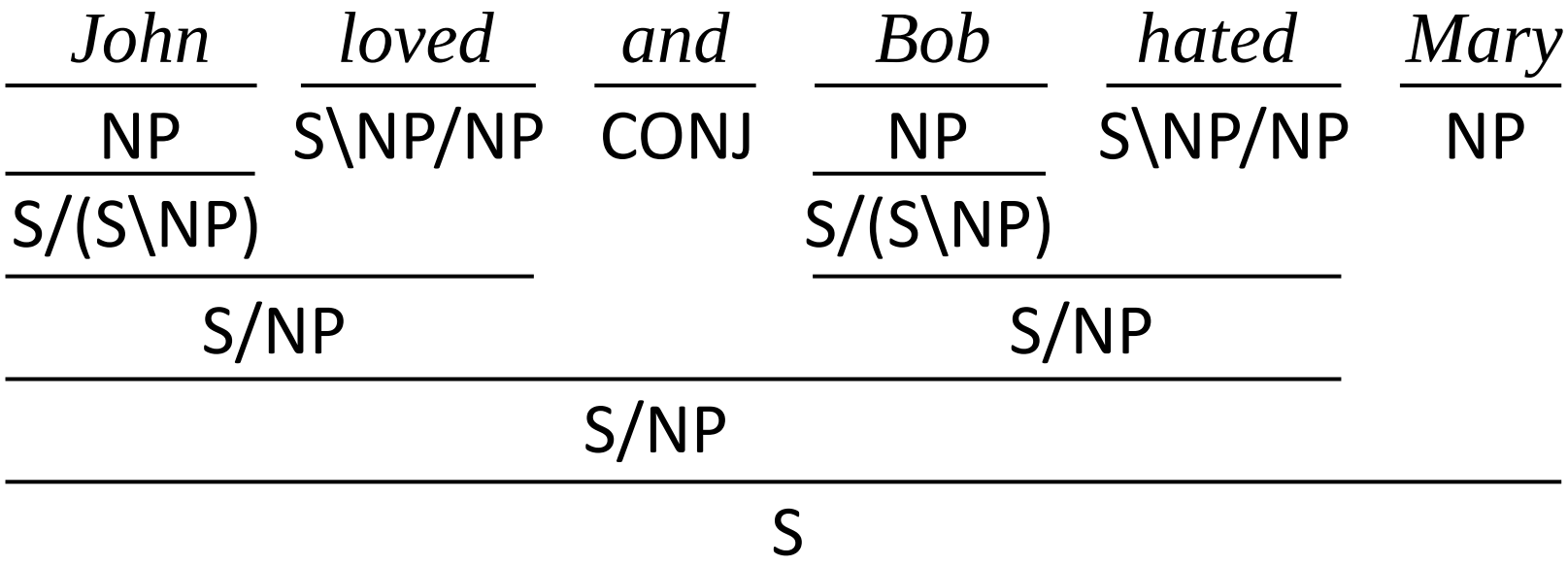
CG

カテゴリ
関数適用規則

CCG

+ 拡張規則
（組合せ規則）

CCG の例



- 関数合成規則
  - X/Y　Y/Z　⇒　X/Z　(>B)
  - Y\Z　X\Y　⇒　X\Z　(<B)
- 型繰上げ規則
  - X　⇒　T/(T\X)　(>T)

主辞駆動句構造文法(HPSG)

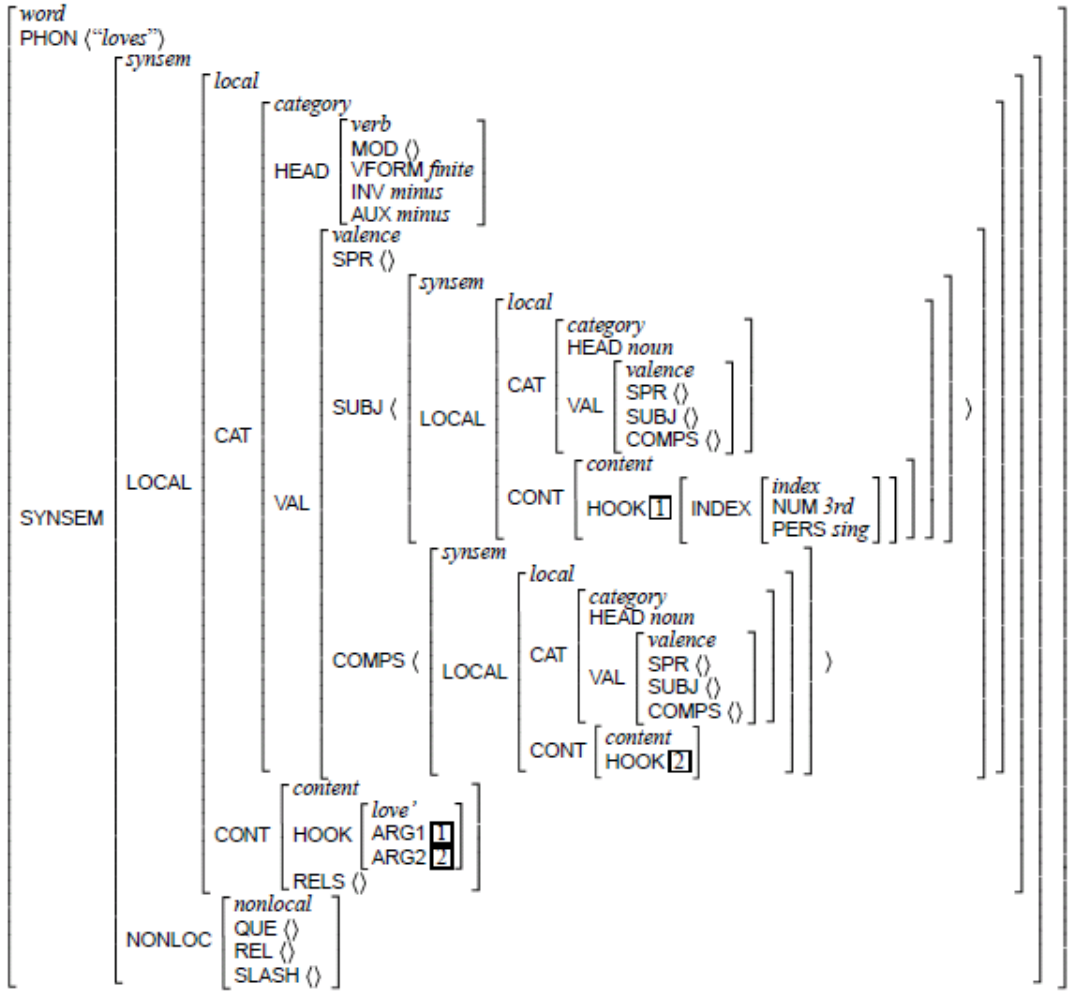
- 文法 ＝ 語彙項目 ＋ 構文規則
  - 語彙項目（≒ 語彙カテゴリ）

$$\begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{SUBJ } \langle \text{NP} \rangle \\ \text{COMPS } \langle \text{NP} \rangle \end{bmatrix} \quad (\fallingdotseq \text{ S\NP/NP})$$

  - 構文規則（≒ 組合せ規則）

$$\boxed{2} \quad \begin{bmatrix} \text{HEAD } \boxed{1} \\ \text{SUBJ } \langle \boxed{2} \rangle \\ \text{COMPS } \langle \rangle \end{bmatrix} \Rightarrow \begin{bmatrix} \text{HEAD } \boxed{1} \\ \text{SUBJ } \langle \rangle \\ \text{COMPS } \langle \rangle \end{bmatrix} \quad (\fallingdotseq \text{ Y X\Y} \Rightarrow \text{X})$$

語彙項目



- 単語の文法的性質を表す
- 単純化した記法で十分

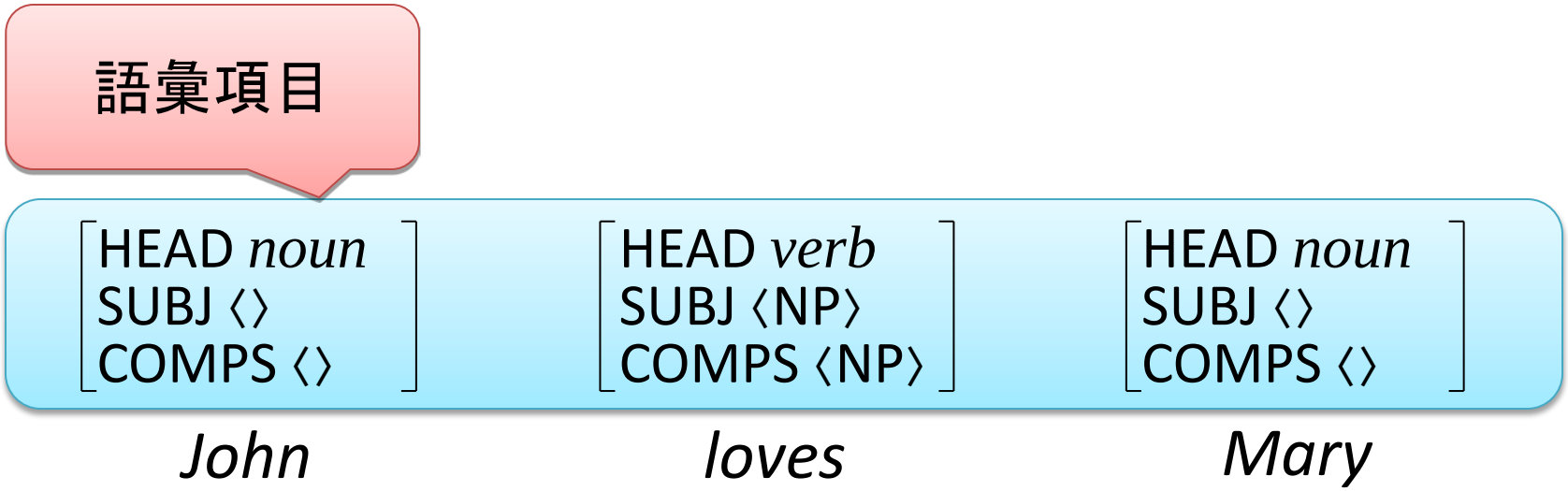
PHON *"loves"*
HEAD *verb*
SUBJ <NP[1] >
COMPS <NP[2] >
CONT *love*([1] , [2])

HEAD *noun*
SUBJ < >
COMPS < >
CONT [1]

NP を主語に取り
NP を目的語に取る
動詞

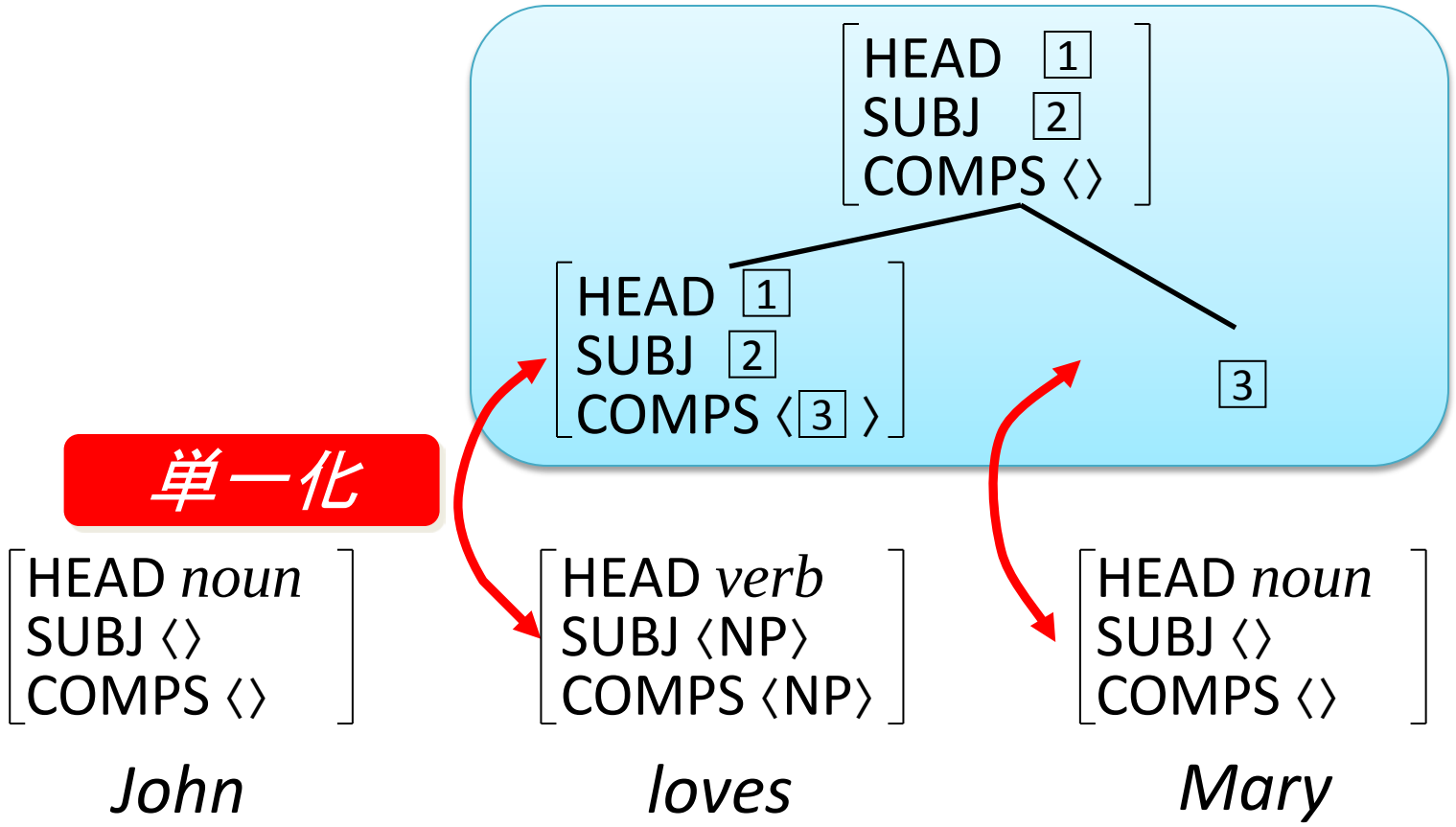
HPSG 構文解析

- 各単語に語彙項目を割り当て



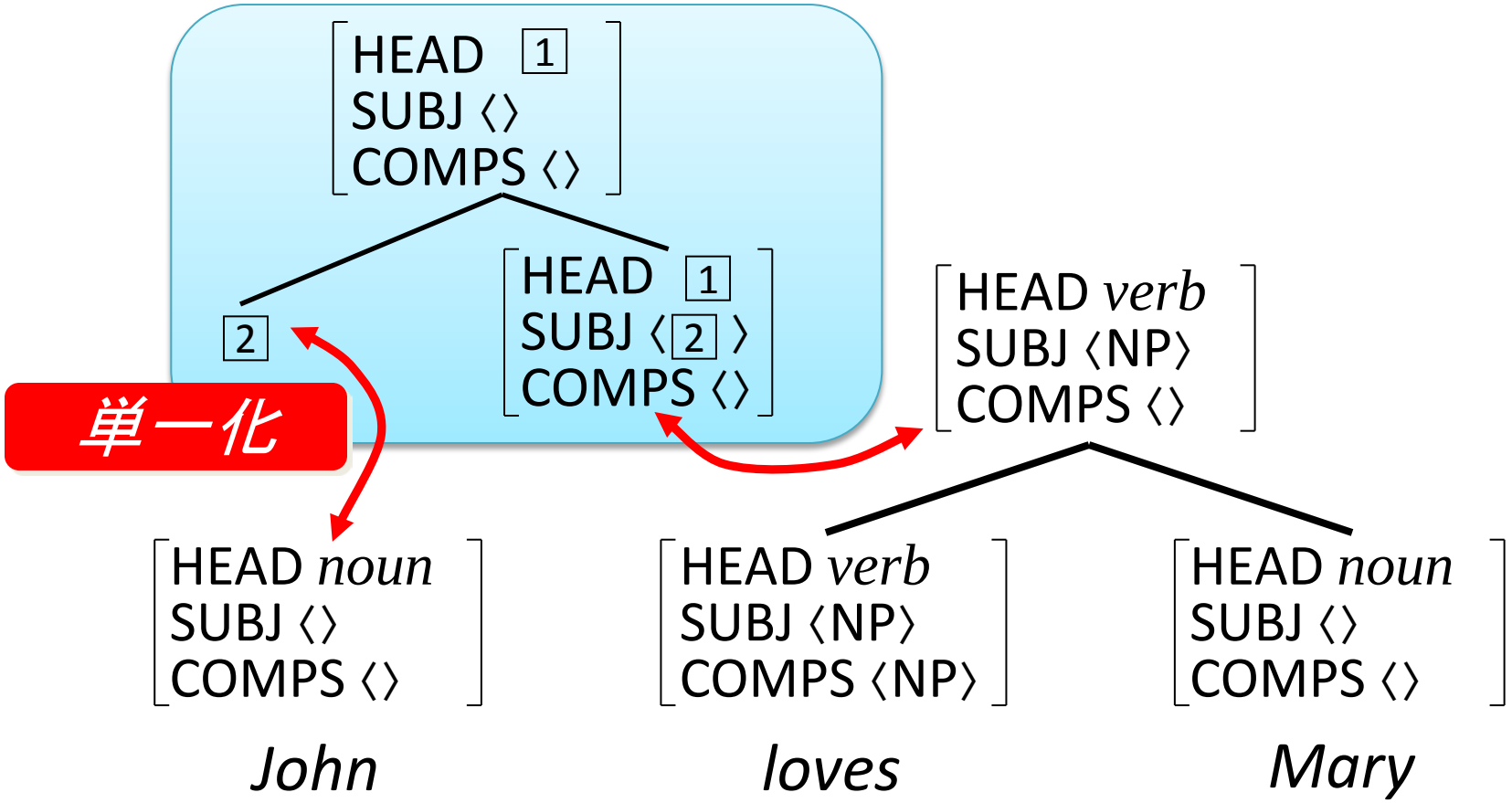
HPSG 構文解析

- 構文規則で句を作る



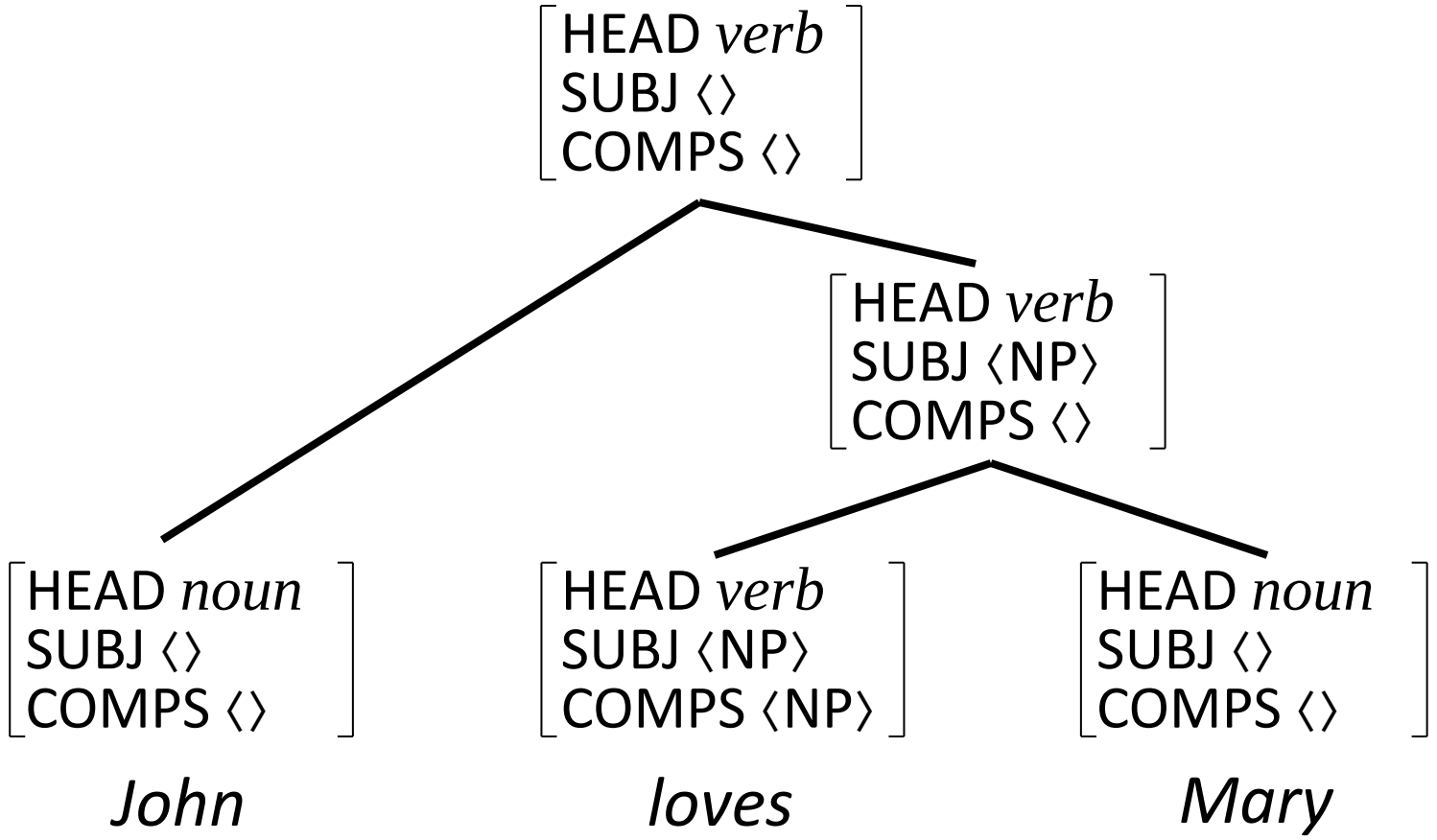
HPSG 構文解析

- 構文規則で句を作る



HPSG 構文解析

- 構文規則で句を作る

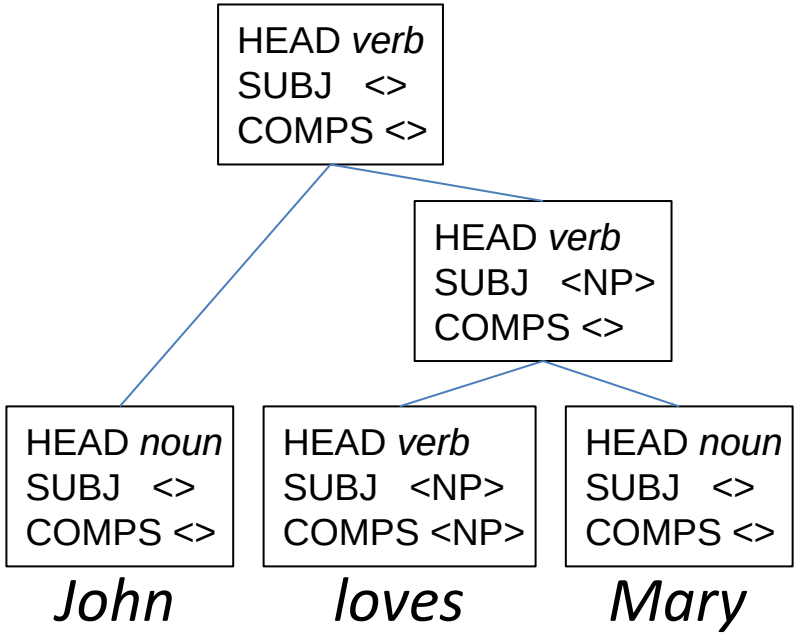


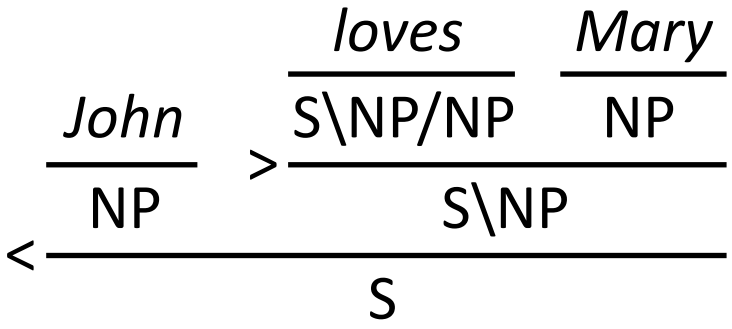
HPSG ≒ CCG

- 基本的な構文の分析はだいたい同じ

  [HEAD *verb*
SUBJ <NP>
COMPS <NP>] ≒ S\NP/NP

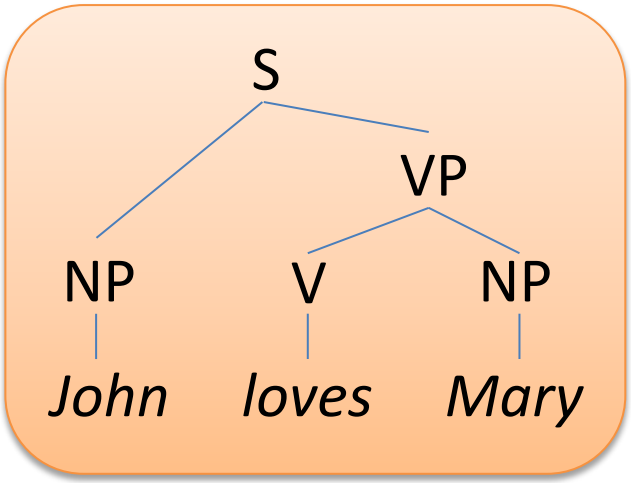
  – もちろん、分析のしかたが異なる言語現象も

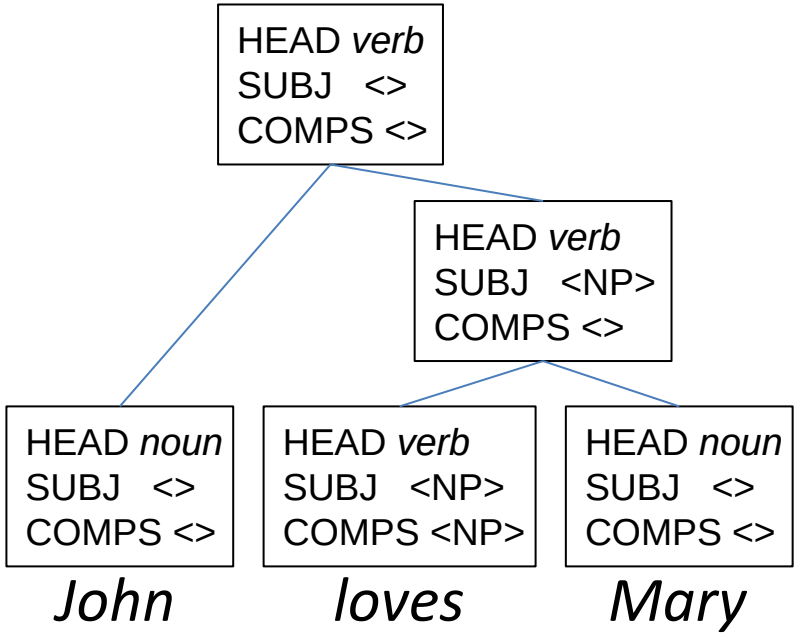


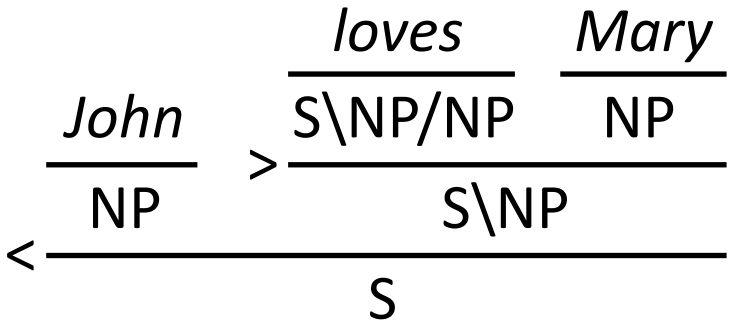


HPSG/CCG 構文解析 ≒ CFG 構文解析

- CFG の解析アルゴリズムがそのまま利用できる
  - CYK法、確率モデル、探索手法、…
- （非）終端記号、生成規則が違う
  - 終端記号＝語彙項目、語彙カテゴリ
  - 生成規則＝構文規則、組合せ規則





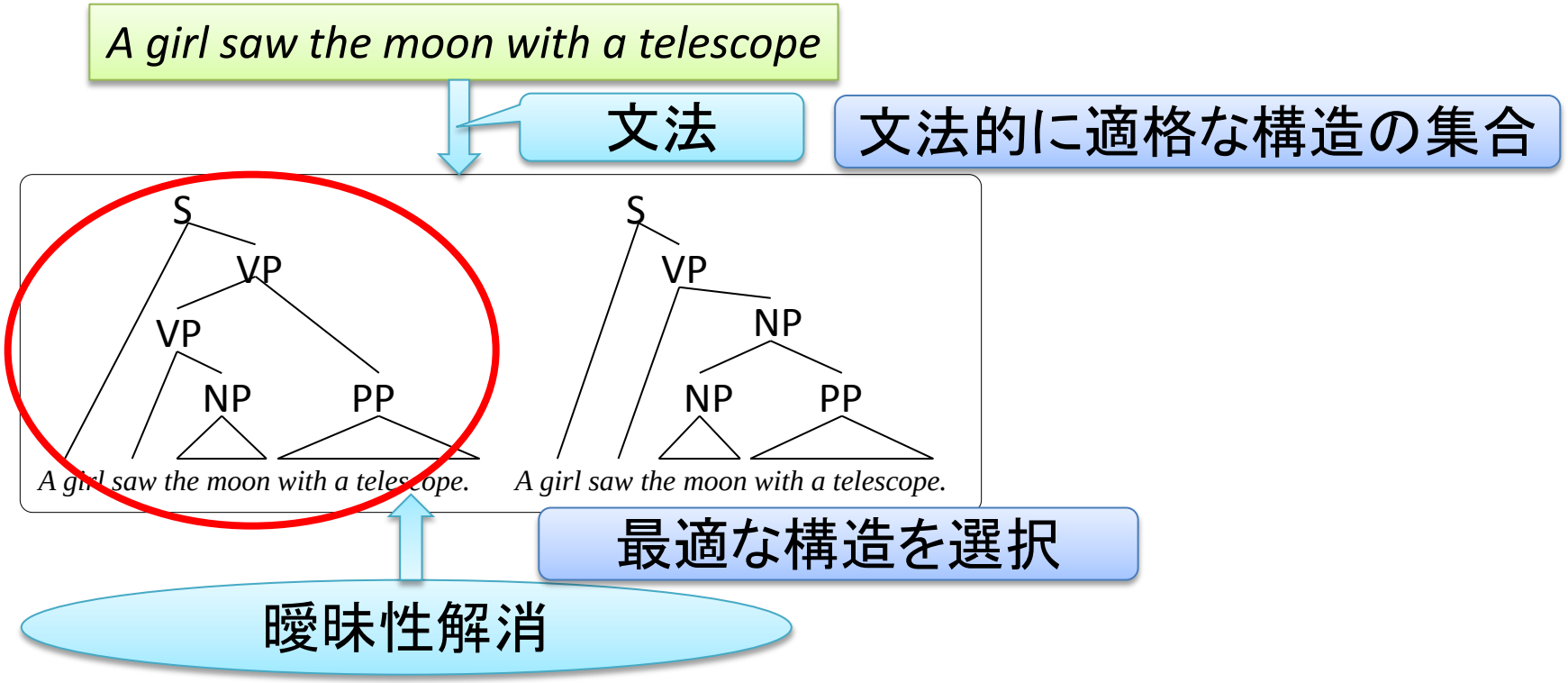


文法と曖昧性解消

- 曖昧性解消を前提とすると、最適解を一つ求めることが目標
  - 全解探索は必要ない
  - 全解探索を前提とした理論やアルゴリズムはあまり重要でない
    - 解析アルゴリズム、計算量、学習可能性、etc.
- 曖昧性解消を前提として、文法と構文解析の関係を考えるべき

文法と曖昧性解消

- 曖昧性解消を前提としたときの文法の役割
  - 解候補の集合を決める
  - 曖昧性解消に有用な構造を提供

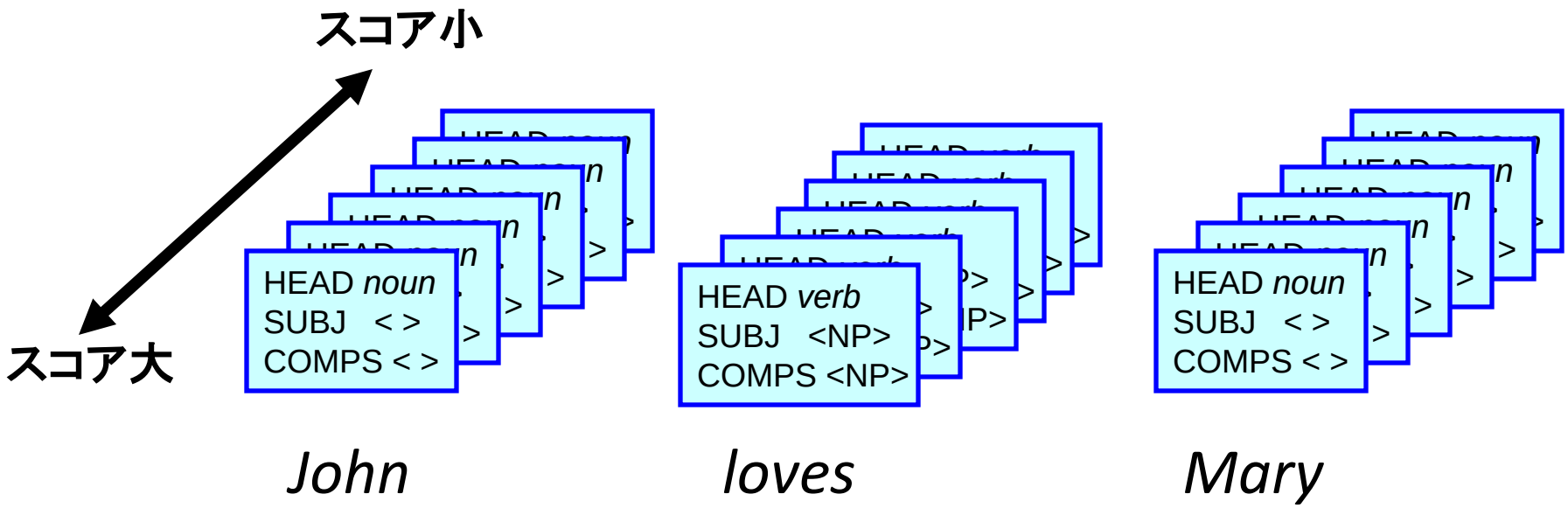


HPSG/CCG と CFG の違い

- 一般論：HPSG や CCG は難しそうに見える（？）
- 曖昧性解消を考えない場合
  - 文（統語構造）の集合が違う
- 曖昧性解消を考える場合
  - 曖昧性解消の単位が違う（後述）
- 計算量
  - 全解探索の計算量：HPSG > CCG > CFG
    - CCG: mildly context-sensitive
    - HPSG: type 0 (?)
  - 実際の解析時間：CFG ≫ HPSG/CCG

Supertagging

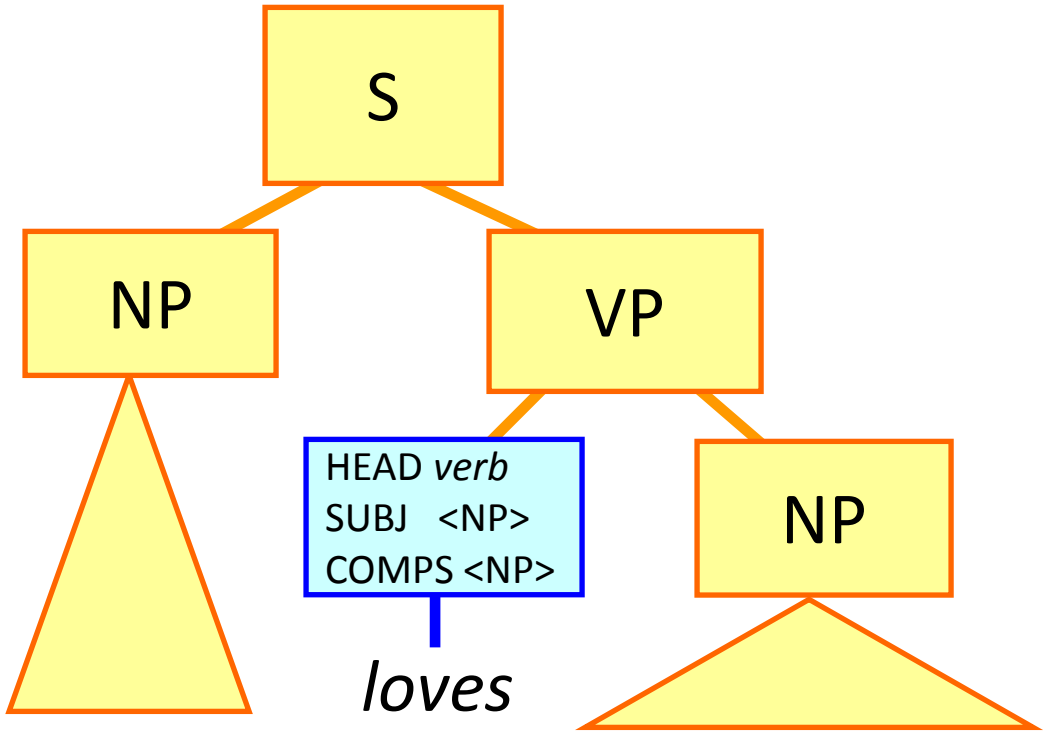
- Supertag = 語彙項目、語彙カテゴリ
- Supertagging = 単語に supertag を割り当てる



Supertagging = “almost parsing”

- Supertag が決まると，その上に作られる統語構造がほぼ決まる

→ 曖昧性のほとんどが supertagging で解消



Supertagging は意外と簡単

- Supertagging は系列ラベリングの一種
  - O(n)
- 簡単な分類器でうまくいく
  - 多くの場合、supertag は局所的情報で決まる

→ 構文解析のほとんどが、単純・高速な supertagging で終わってしまう

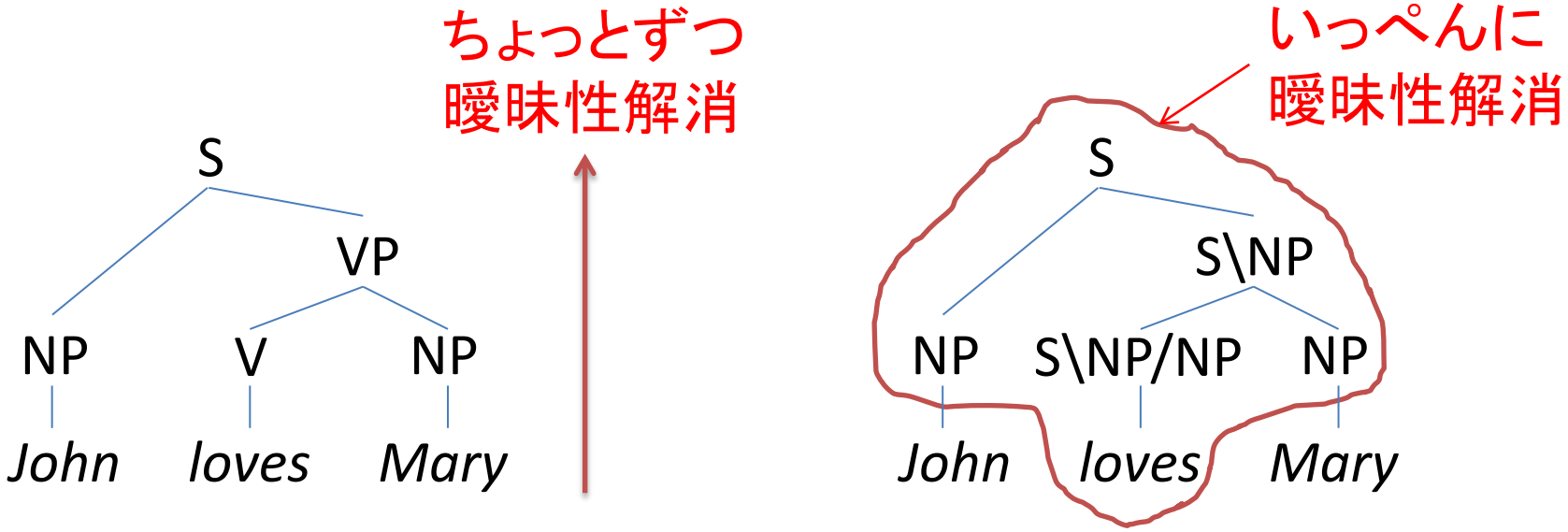
おそらく目的語コントロール動詞

... man forced his friend to ...

... NN VBD PRP$ NN TO ...

文法と曖昧性解消

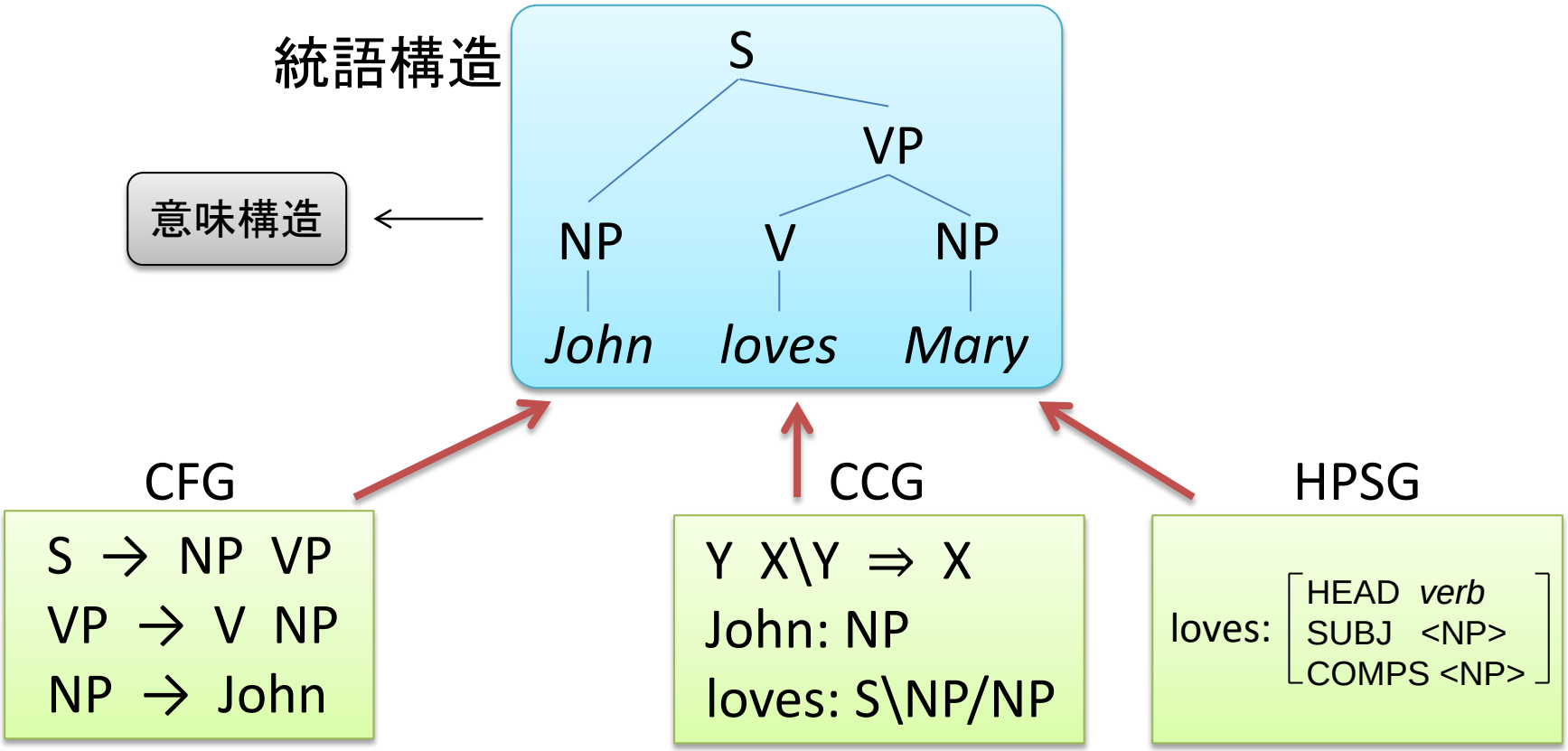
- CFG構文解析とは、曖昧性解消の単位が違う



- 現在の構文解析技術の多くが同様のアイディアに基づいている
  - Lexicalization, head percolation, symbol annotation…

文法と計算プロセス

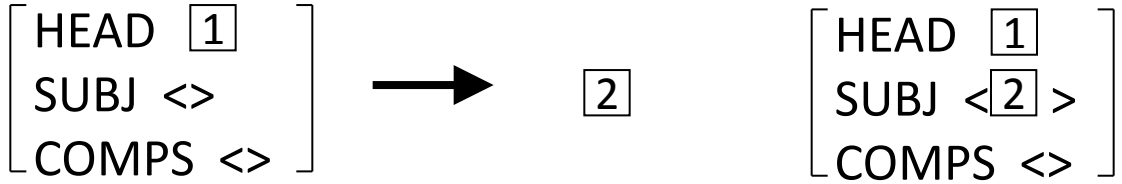
- 統語構造＝構文解析の計算結果
- CFG と HPSG/CCG の違いは、計算方法（プロセス）の違いと見ることもできる



HPSG 構文規則

- HPSG の構文規則は、類似したCFG規則を一般化したもの

主語 + 主辞



主辞 + 目的語･補語

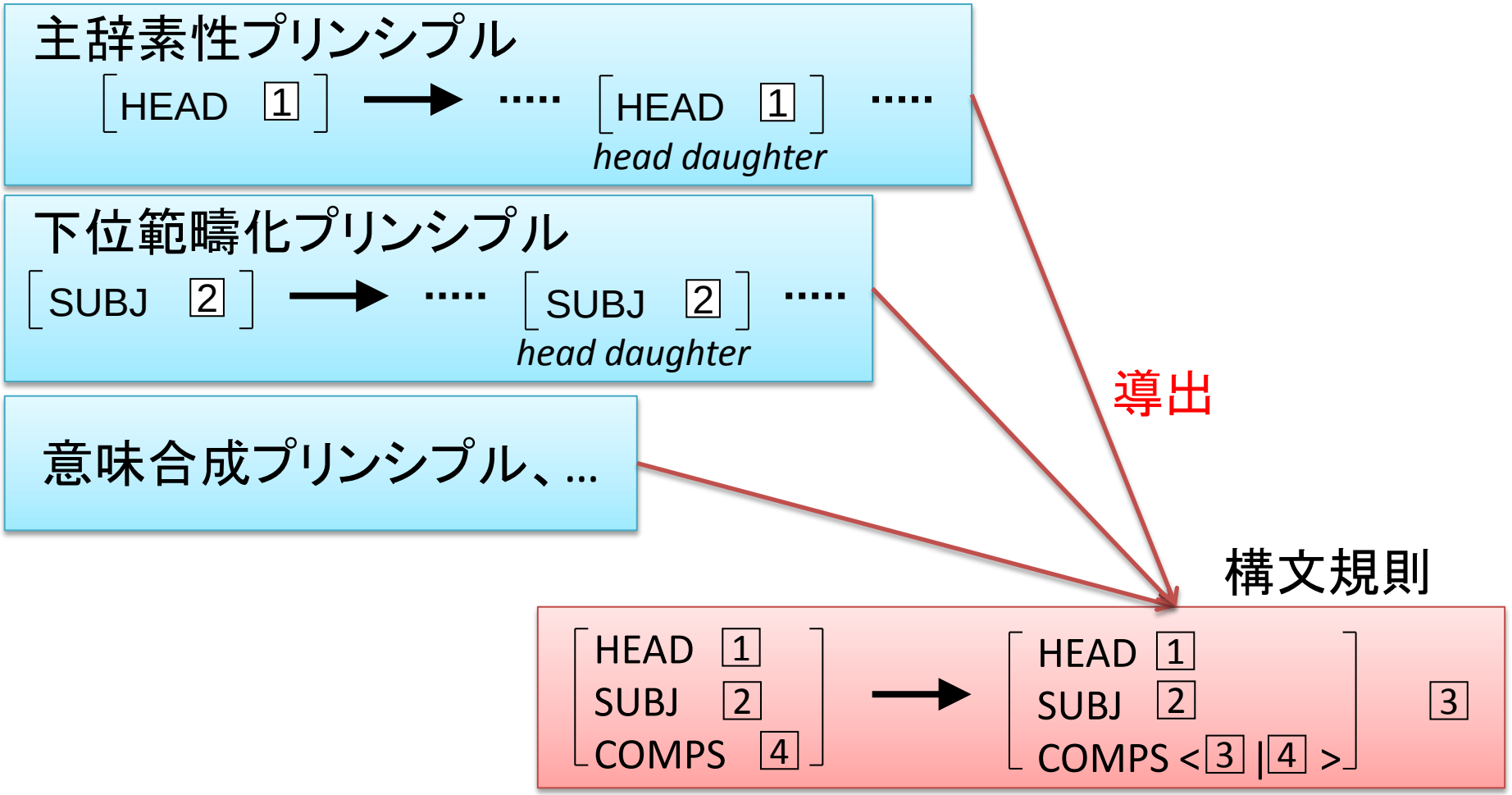
プリンシプル＝文法の文法

- 構文規則や語彙項目は、それ自身がプリンシプルから導出される
- 主辞素性プリンシプル
  - HEADの値は主辞から受け継がれる

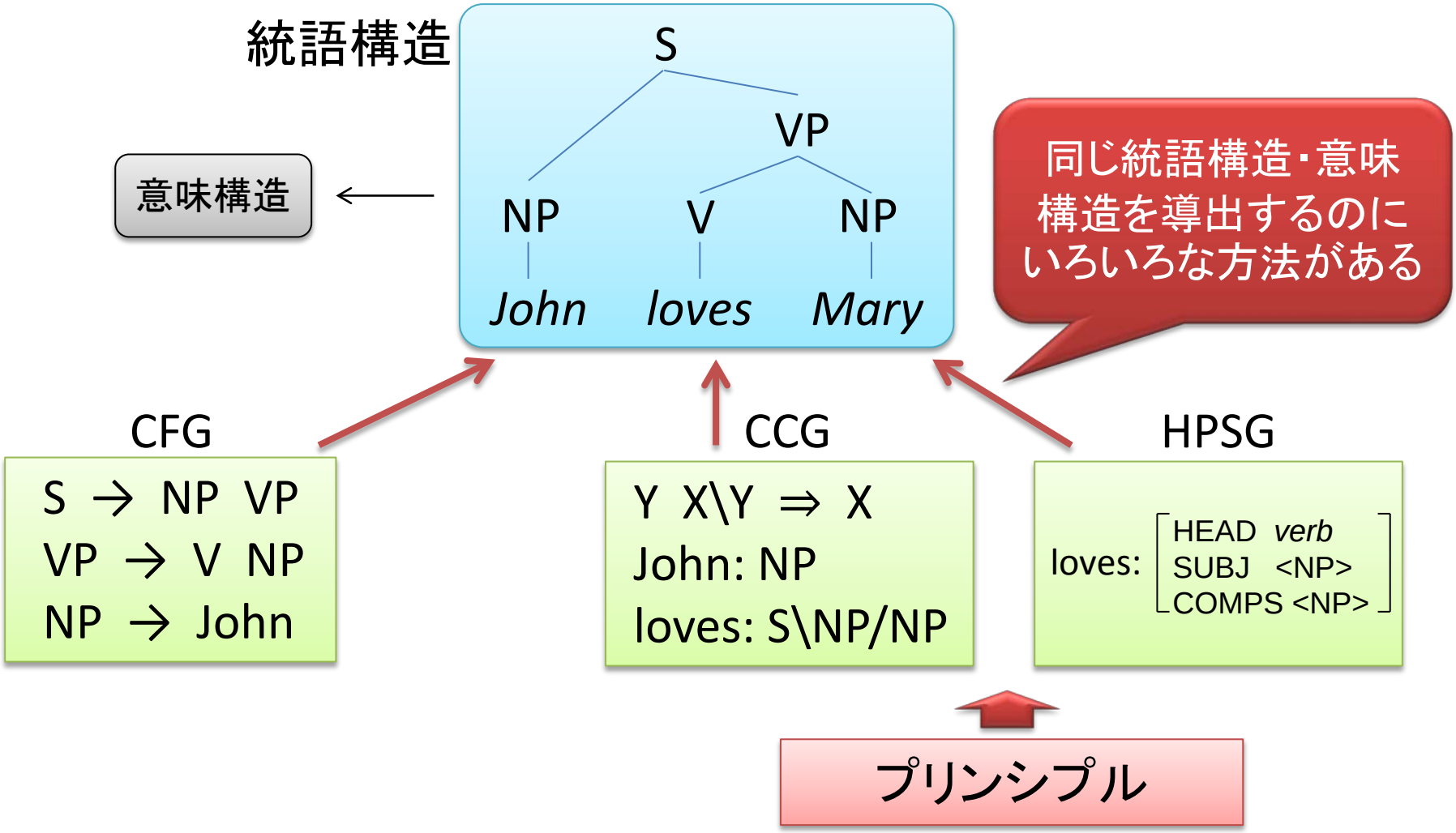
- 下位範疇化プリンシプル
- 意味合成プリンシプル
- 語彙規則
- …

プリンシプル＝文法の文法

- プリンシプルから文法を導出する



いろいろな計算プロセス（＝分解・一般化）の可能性



文法と計算プロセス

- CFG と HPSG/CCG の違いは、統語構造に至る計算方法（プロセス）の違い
  - HPSG と CCG の間にも重要な違いがある（割愛）

→学習の観点から見ると、統語構造の分解（一般化）のしかたが違う

- 文法に分解（一般化）した上で、さらに一般化した構造を探求することもできる ＝言語理論
  - cf. Grammar Matrix

おわりに

- 構文解析と文法、言語理論、曖昧性解消の関係について
- 自然言語処理における現状
  - 現在の構文解析手法は、言語理論が提示する一般化・構造（＝モデル）まで達していない
  - 構文解析に本当に有効な一般化・構造は、言語理論のそれと一致するとは限らない
- 統語構造・意味構造を高精度、高速かつシンプルに計算できる「原理」を探したい