人工知能特論II　第3回

二宮　崇

今日の講義の予定

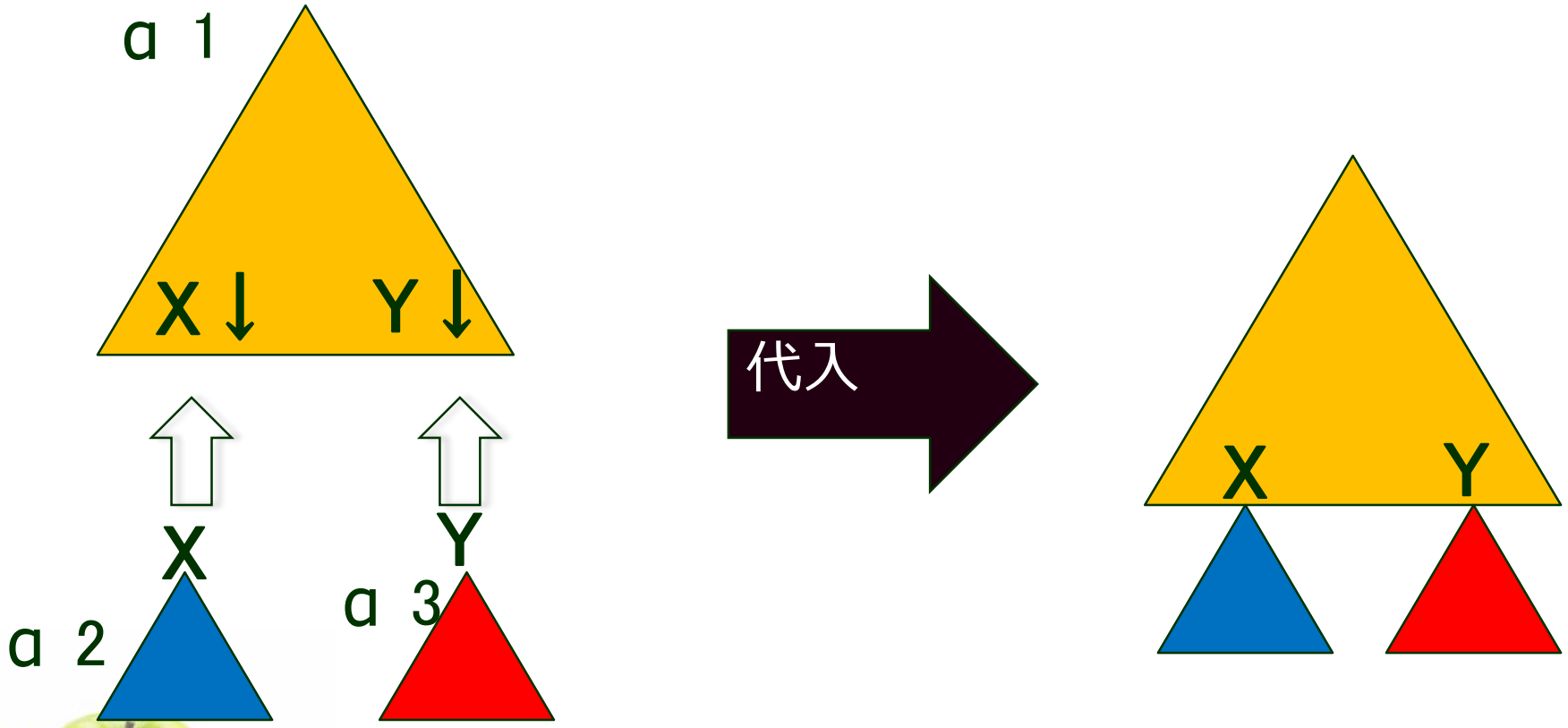
- TAG (Tree Adjoining Grammar)
- Dependency Grammar (依存文法)

TAG

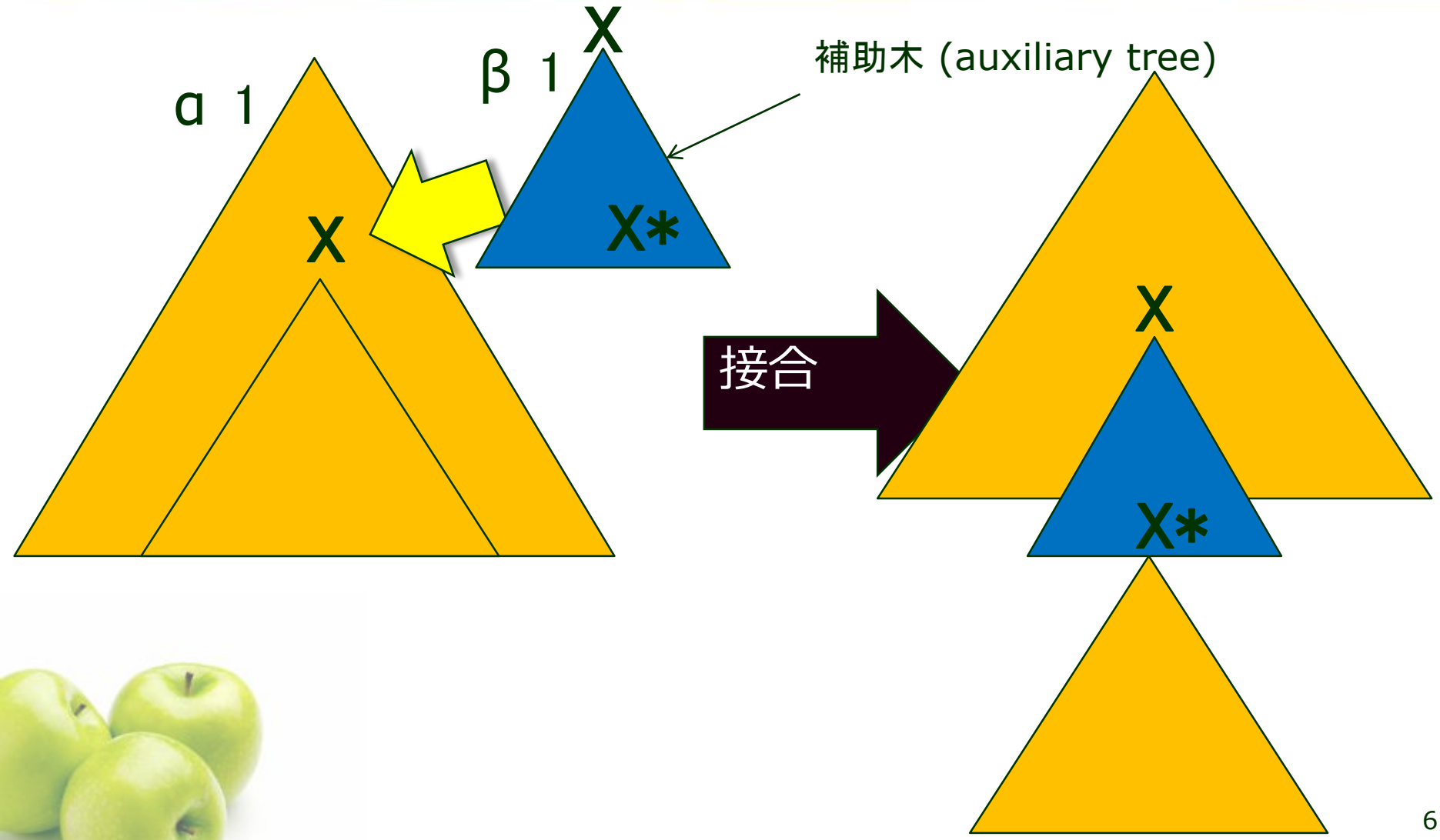
TAG (Tree Adjoining Grammar, 木接合文法)

- TAG
  - A. K. Joshiによって提案
  - 複雑な形をした句構造木そのものが文法の基本要素 (基本木, elementary trees)
    - 初期木(initial trees)
    - 補助木(auxiliary trees)
  - 2種類の操作で構文解析
    - 代入(substitution)
    - 接合 (adjoining)

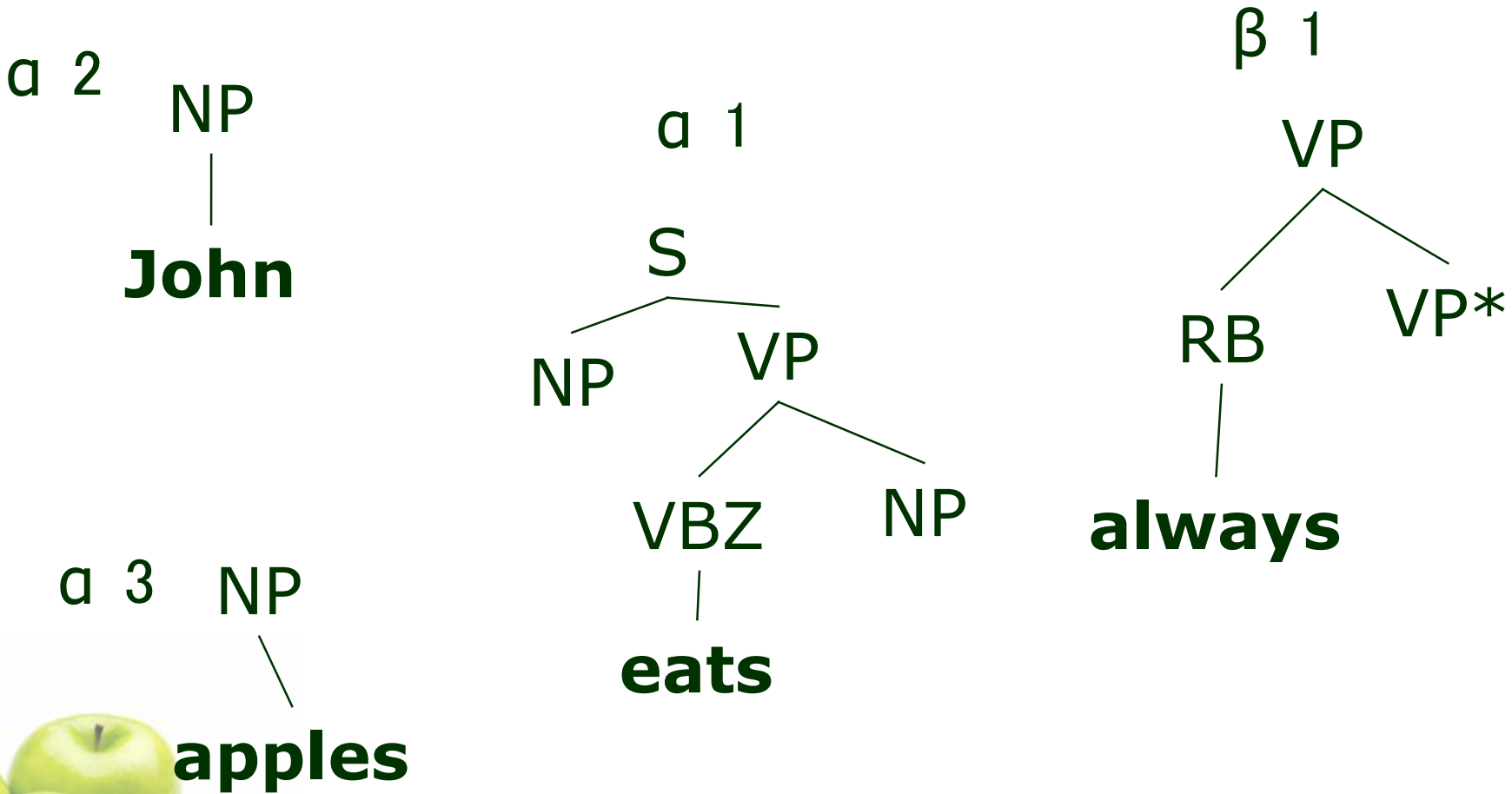
TAG：代入 (substitution)



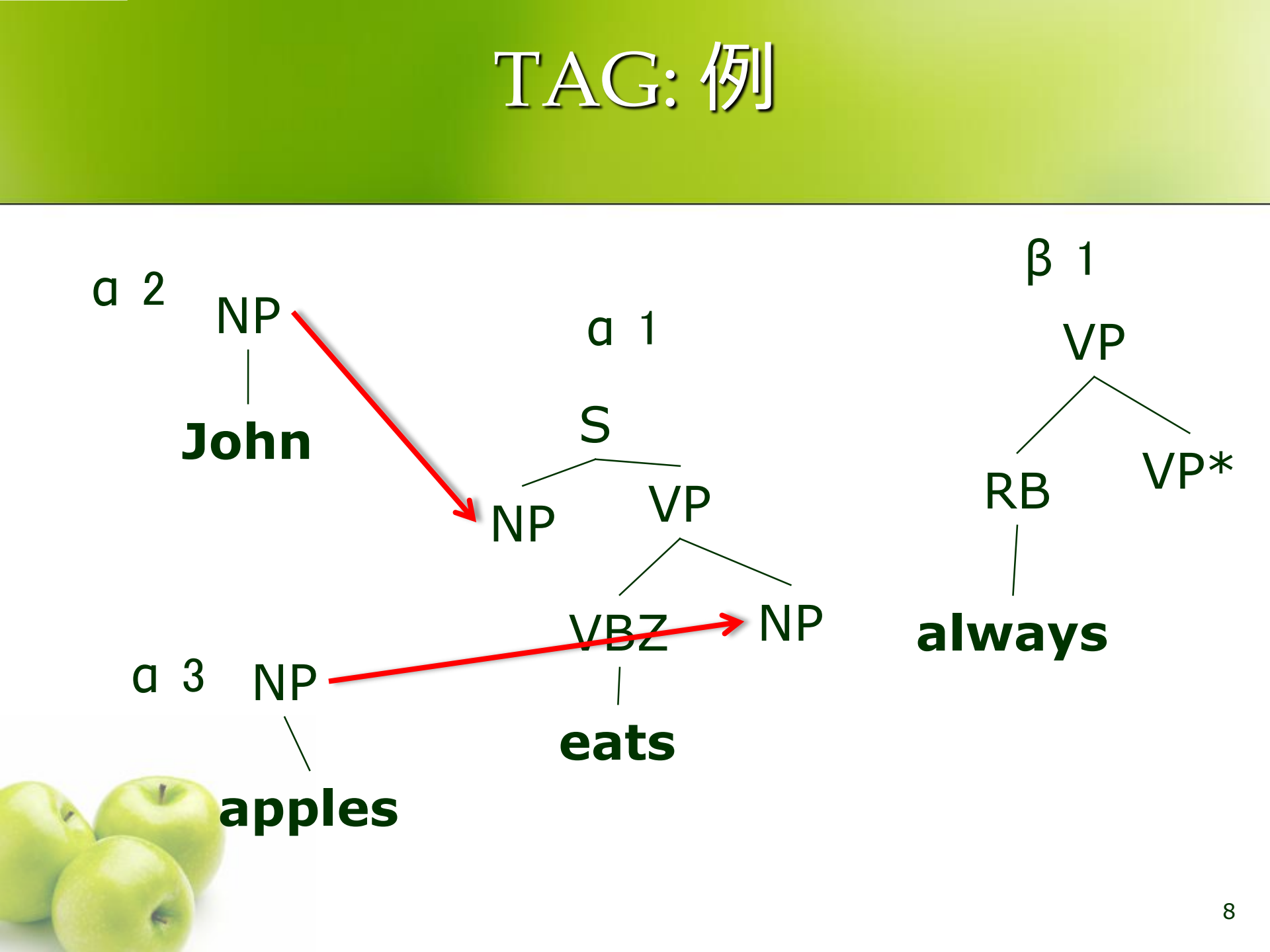
TAG：接合 (adjoining)



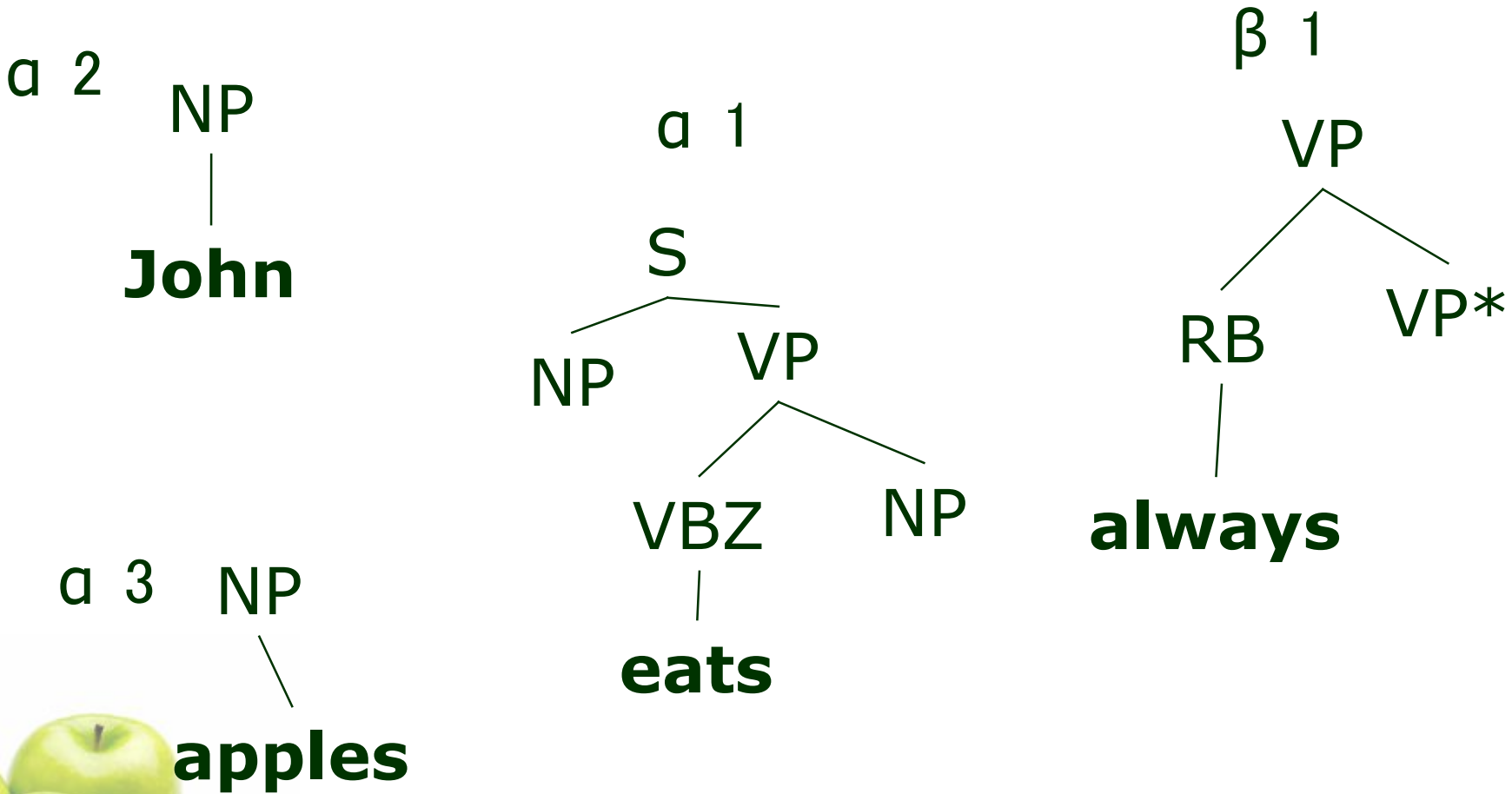
TAG: 例



TAG: 例



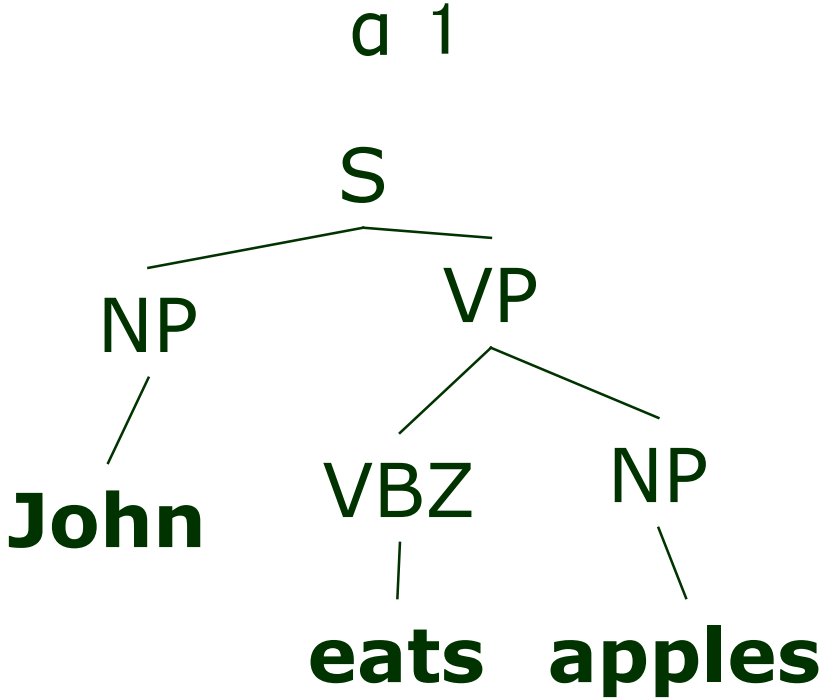
TAG: 例



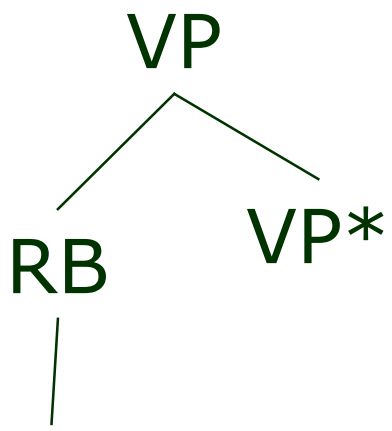
TAG: 例

α 2

α 1

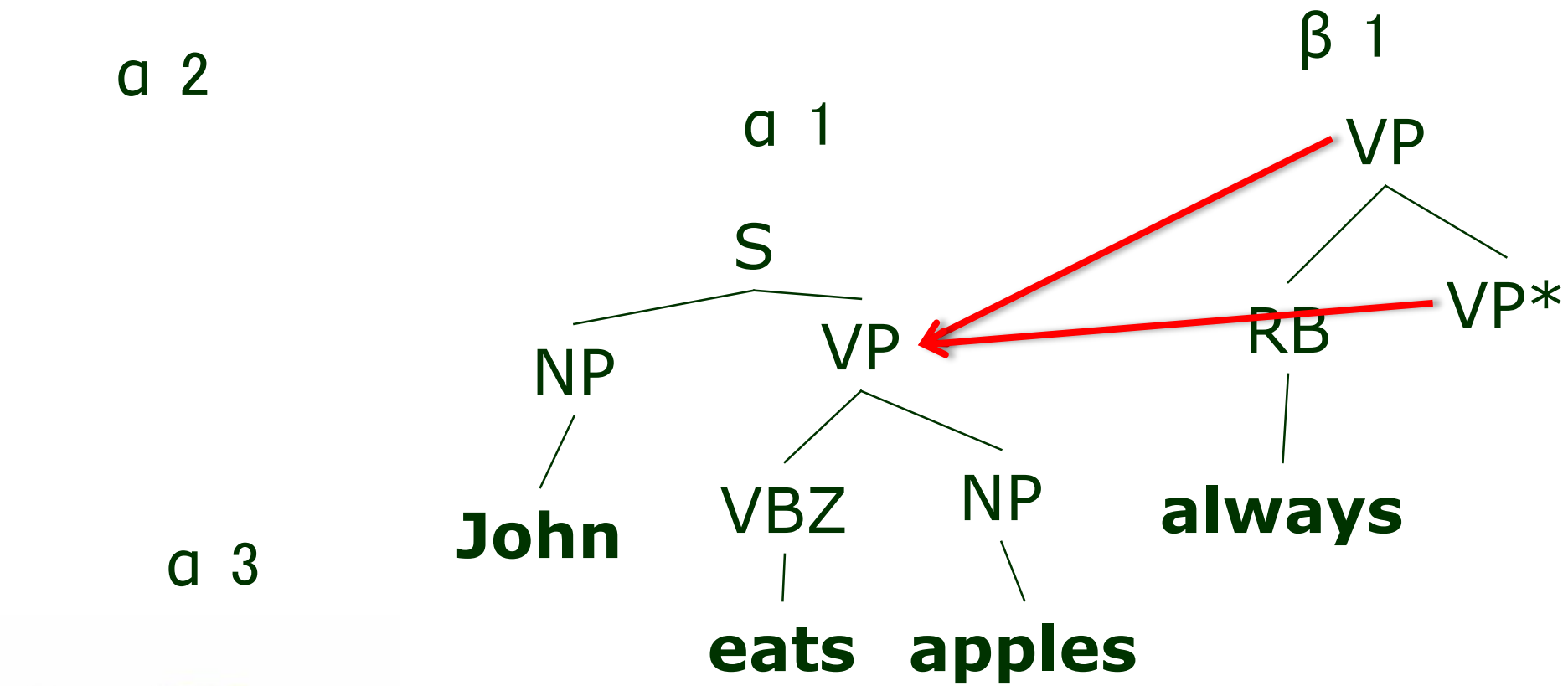


β 1



α 3

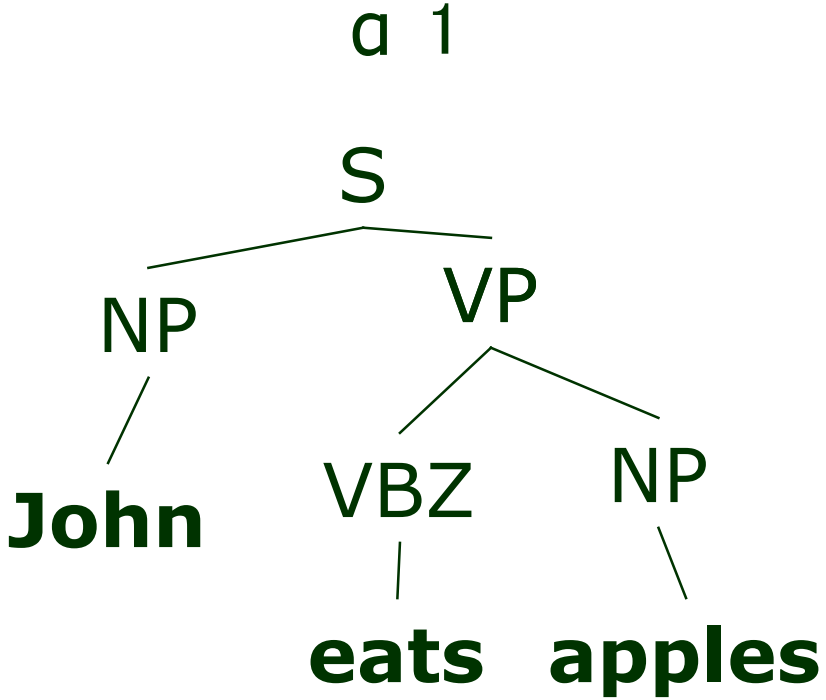
TAG: 例



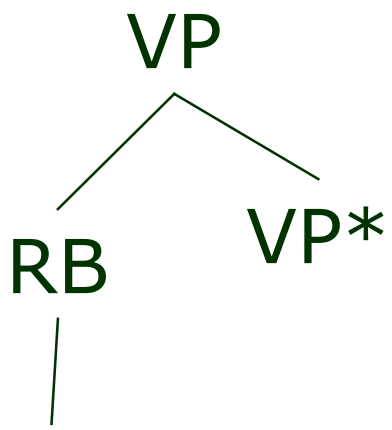
TAG: 例

α 2

α 1



β 1



α 3

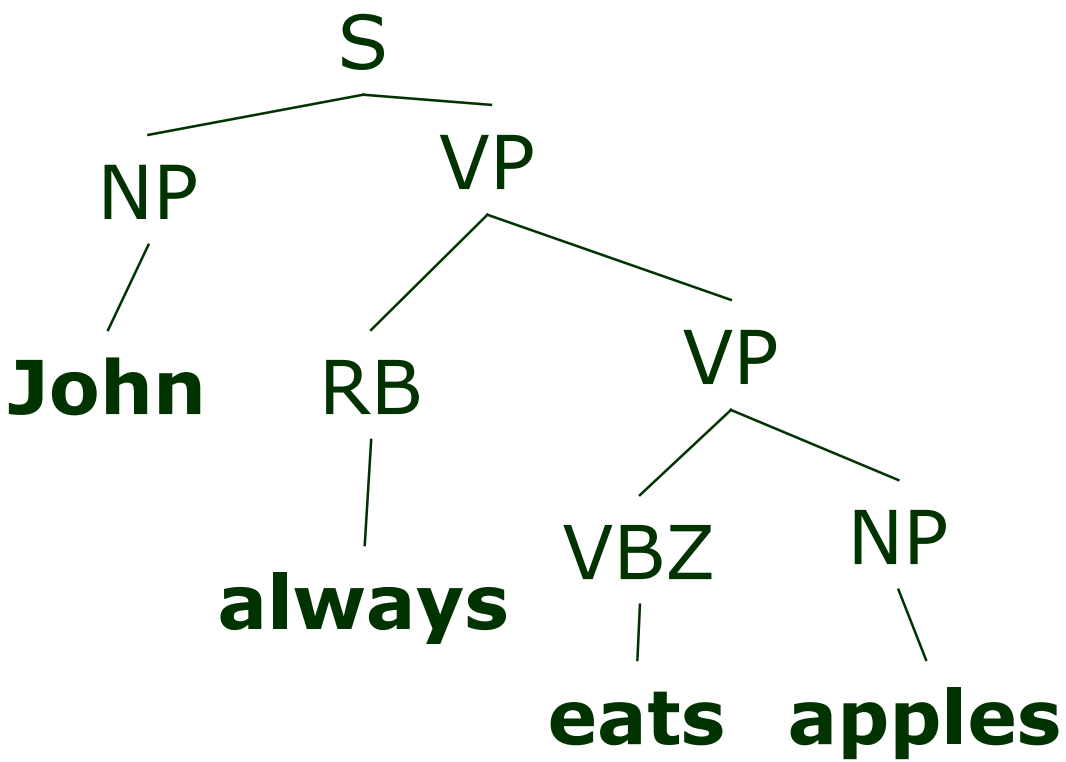
TAG: 例

α 2

α 1

β 1

α 3

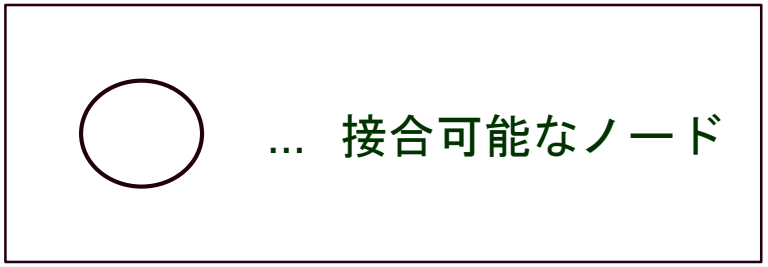


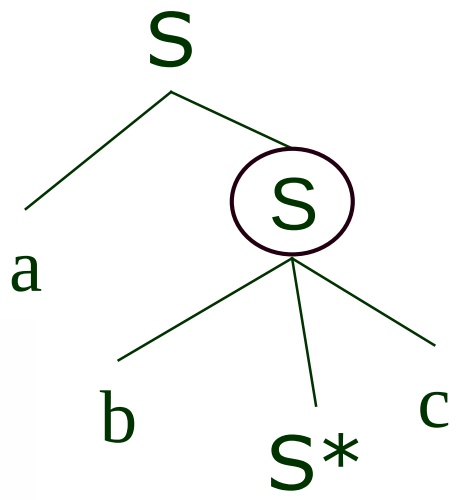
TAG

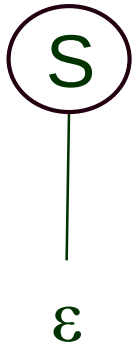
- TAGは形式的に(∑, NT, I, A, S)の5つ組
  - ∑ … 終端記号の集合
  - NT … 非終端記号の集合
  - I … 初期木の集合
  - A … 補助木の集合
  - S … S ⊂NT, 文を表す非終端記号
  - 接合可能なノードを指定しても良い
- 弱文脈依存文法 (mildly context sensitive grammar)
- 導出過程木が依存構造に対応

TAGの生成能力

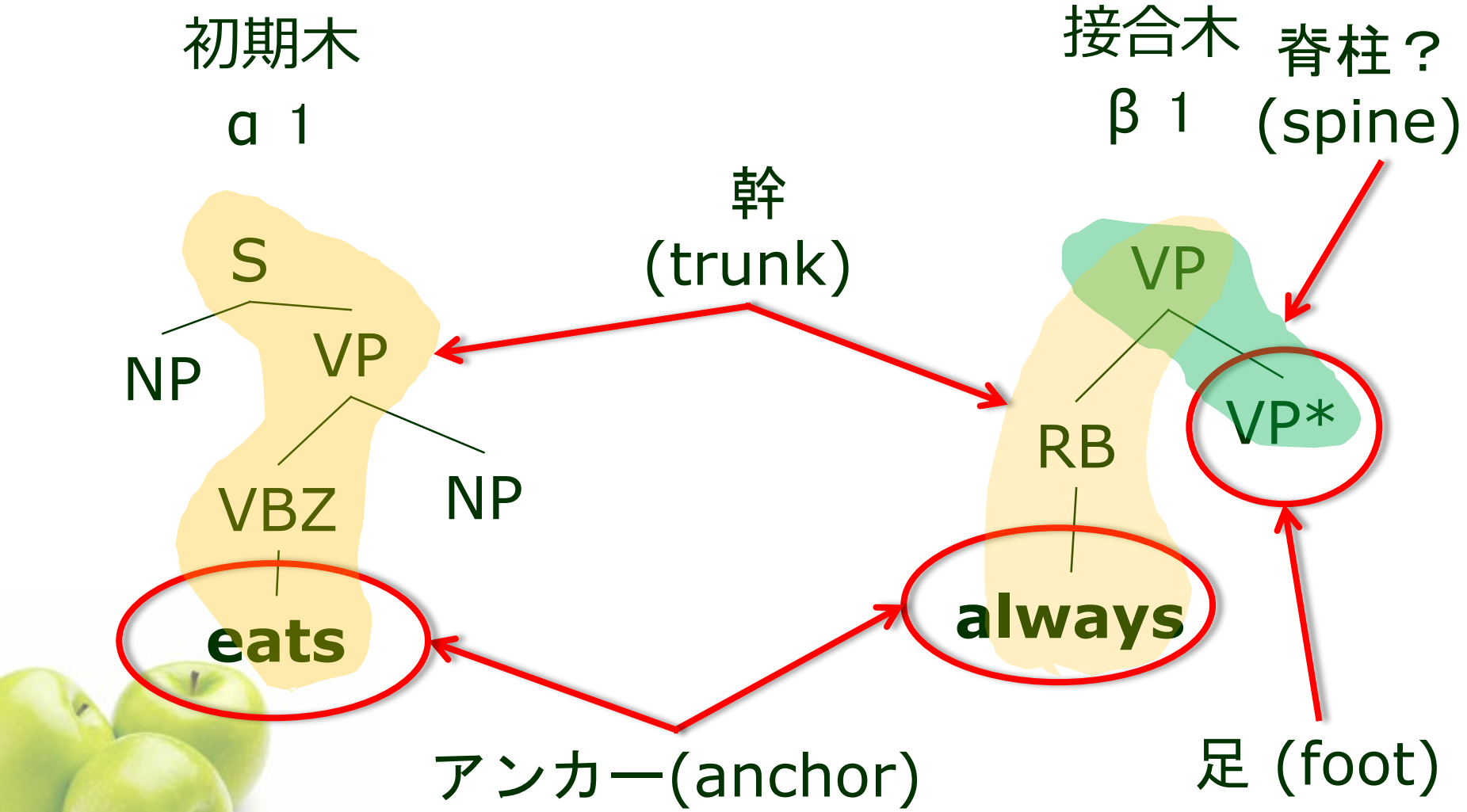
- CFGでは表現できないがTAGでは表現できる言語
  - $L=\{a^n b^n c^n \mid n \in \mathbb{N}\}$



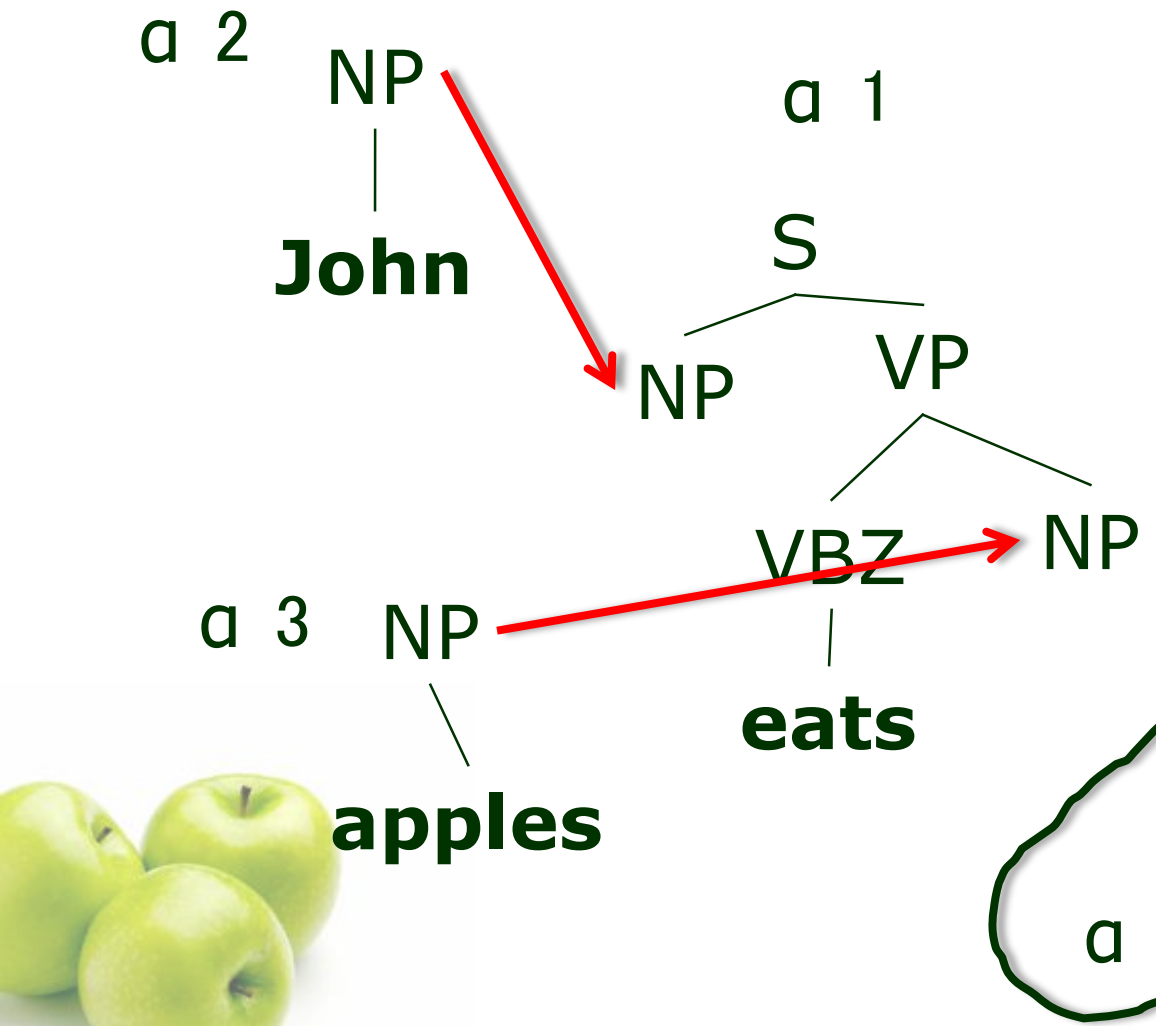


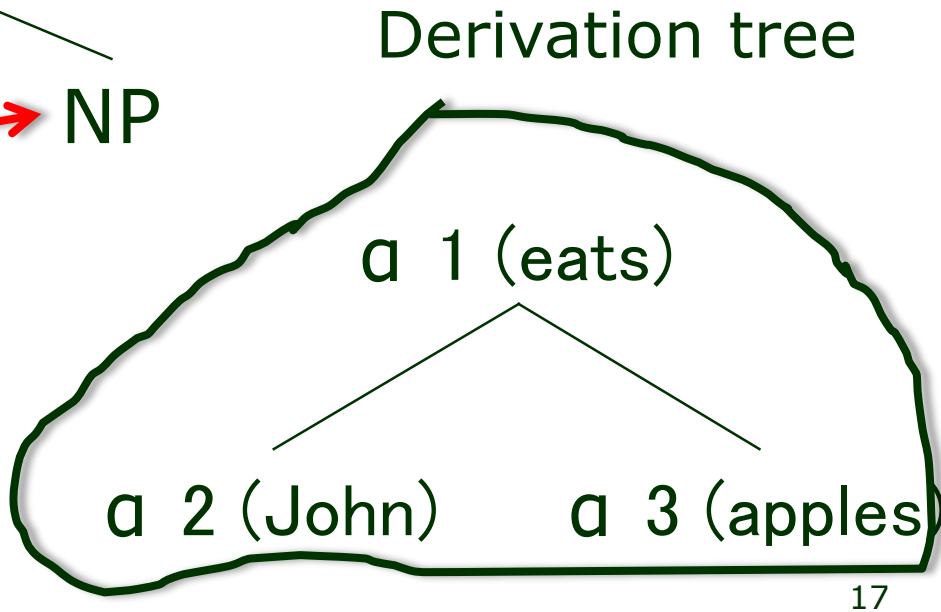


TAG: 初期木と接合木

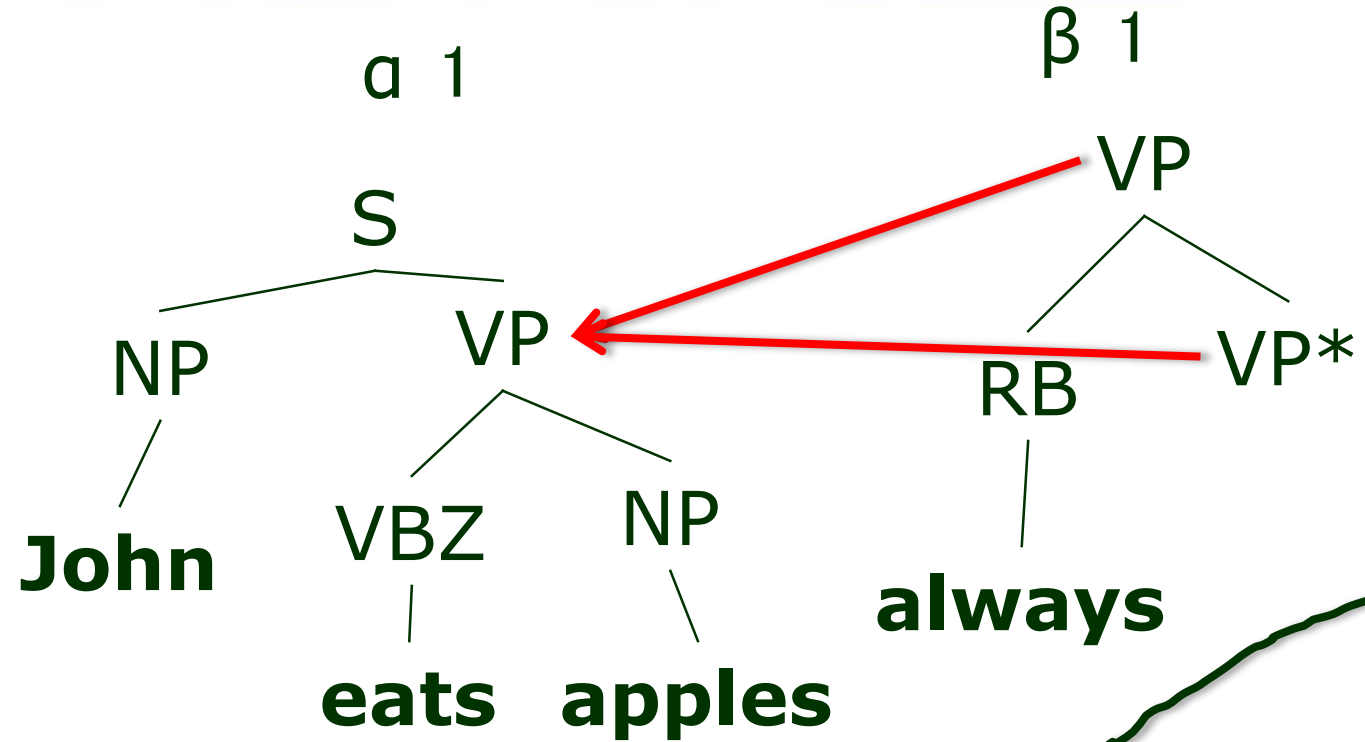

初期木
α 1
接合木
β 1
脊柱？
(spine)
幹
(trunk)
S
NP
VP
VBZ
NP
eats
VP
RB
VP*
always
アンカー(anchor)
足 (foot)

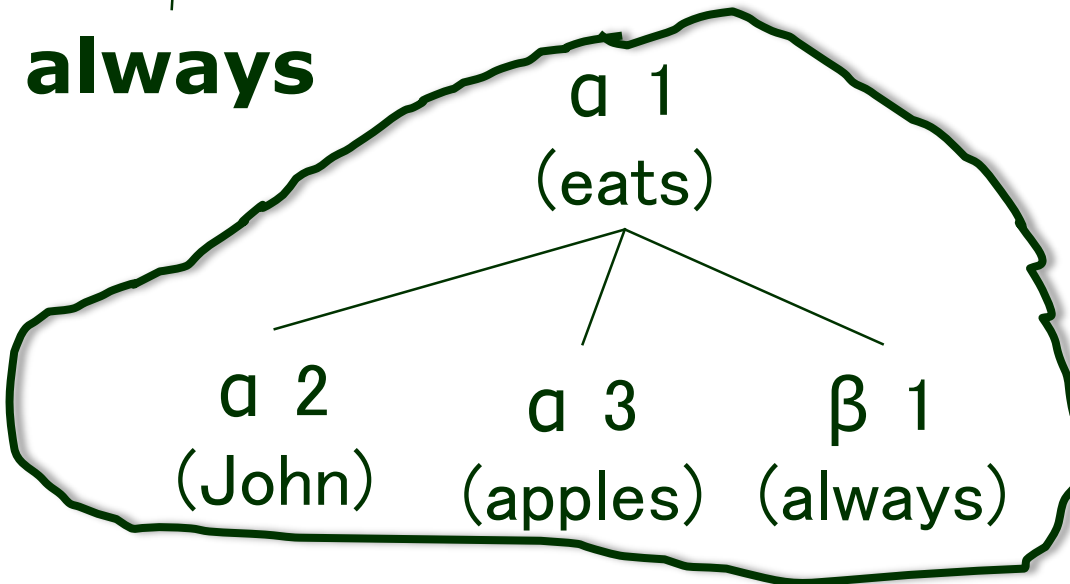
TAG: 導出過程木 (derivation tree)



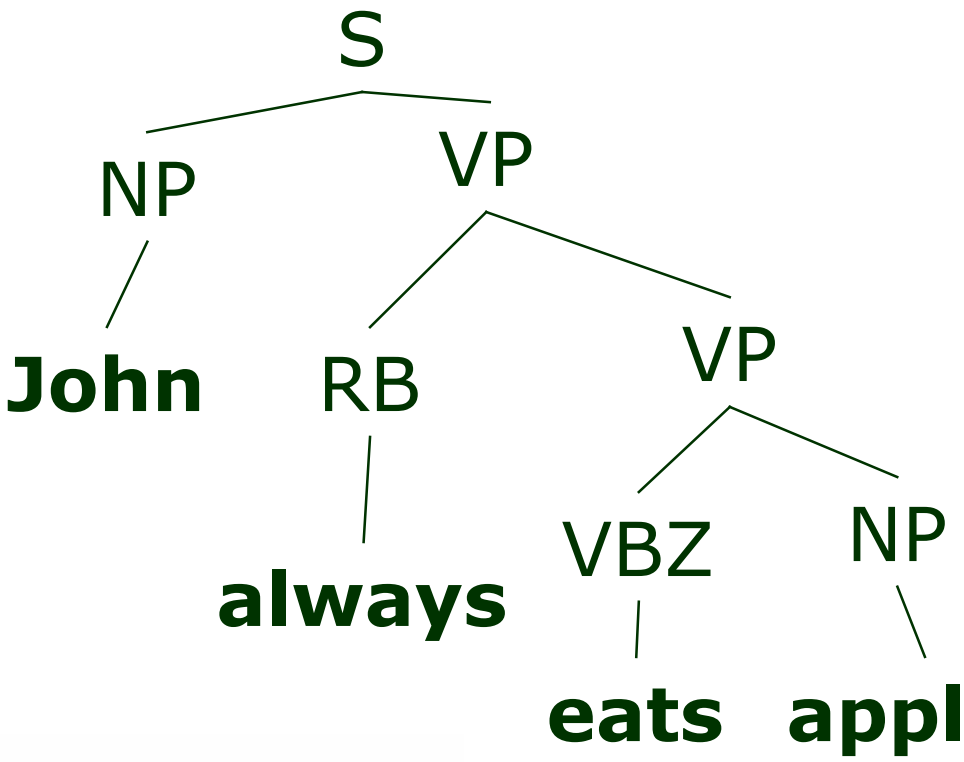


TAG: 導出過程木 (derivation tree)

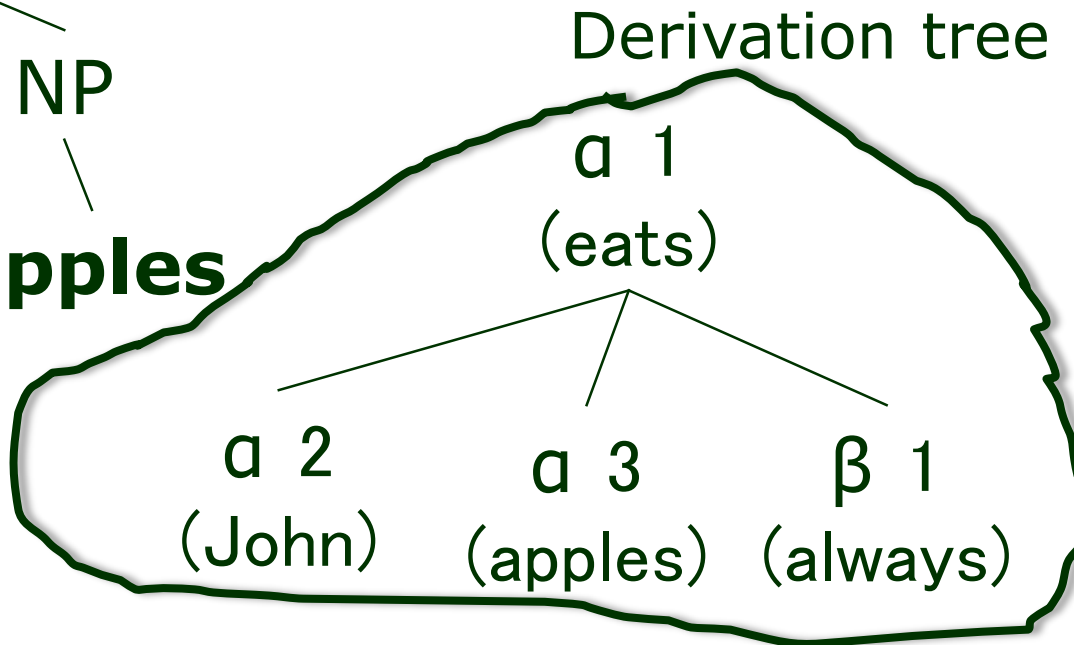




CFG v.s. TAG



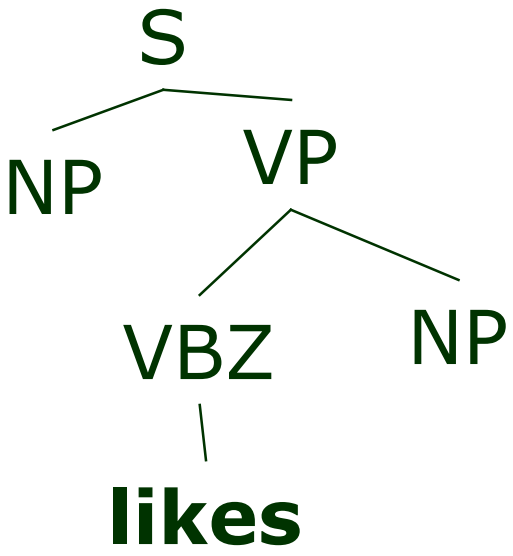
CFGだと、eatsの主語がJohnであることがわかりにくい



TAGだと、eatsの主語がJohnであることがすぐわかる

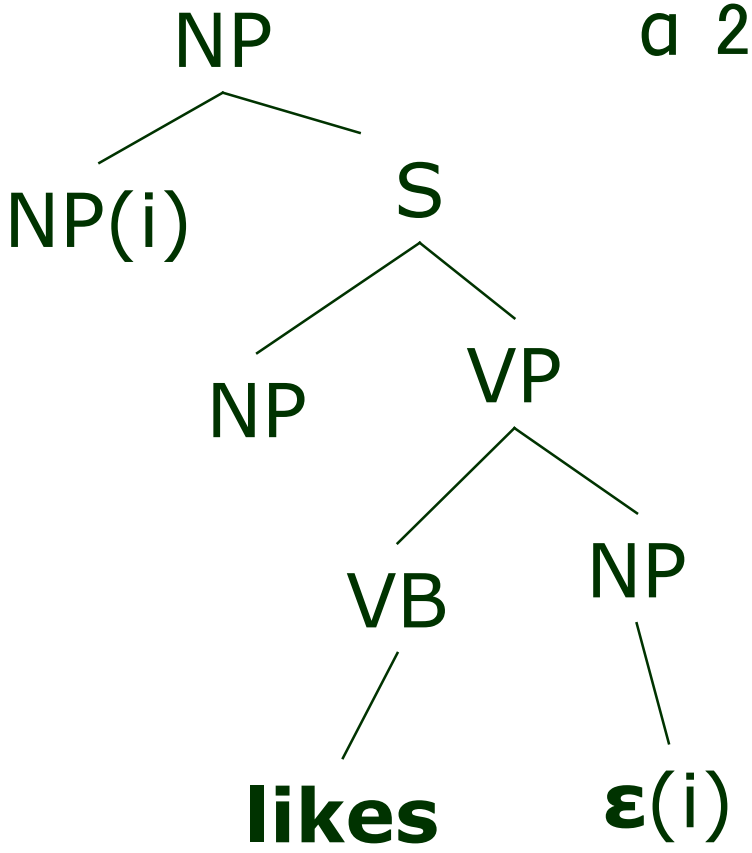
Non-local Dependencies

α 1



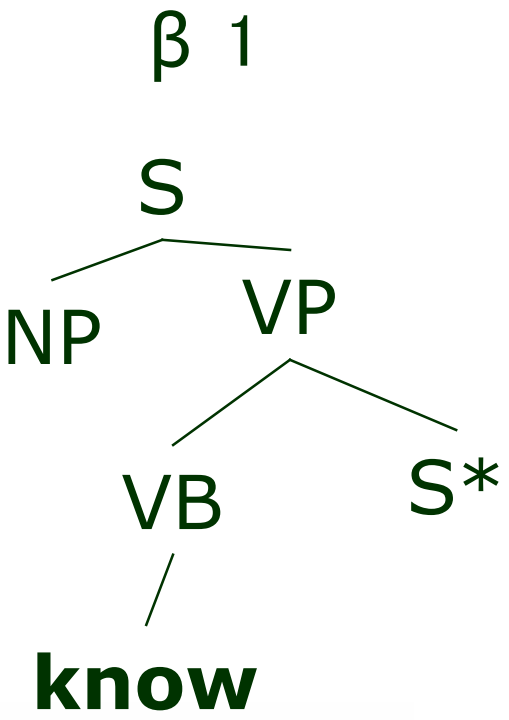
他動詞

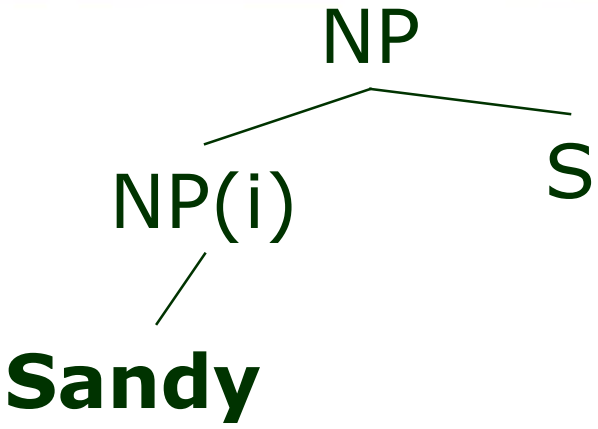
α 2



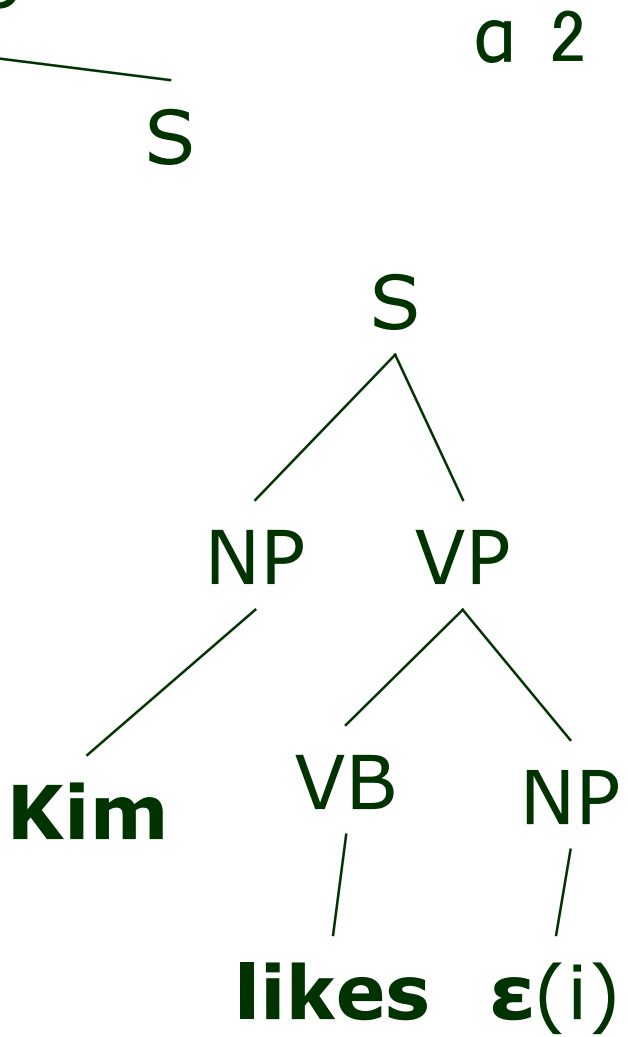
目的語が外置された他動詞

Non-local Dependencies

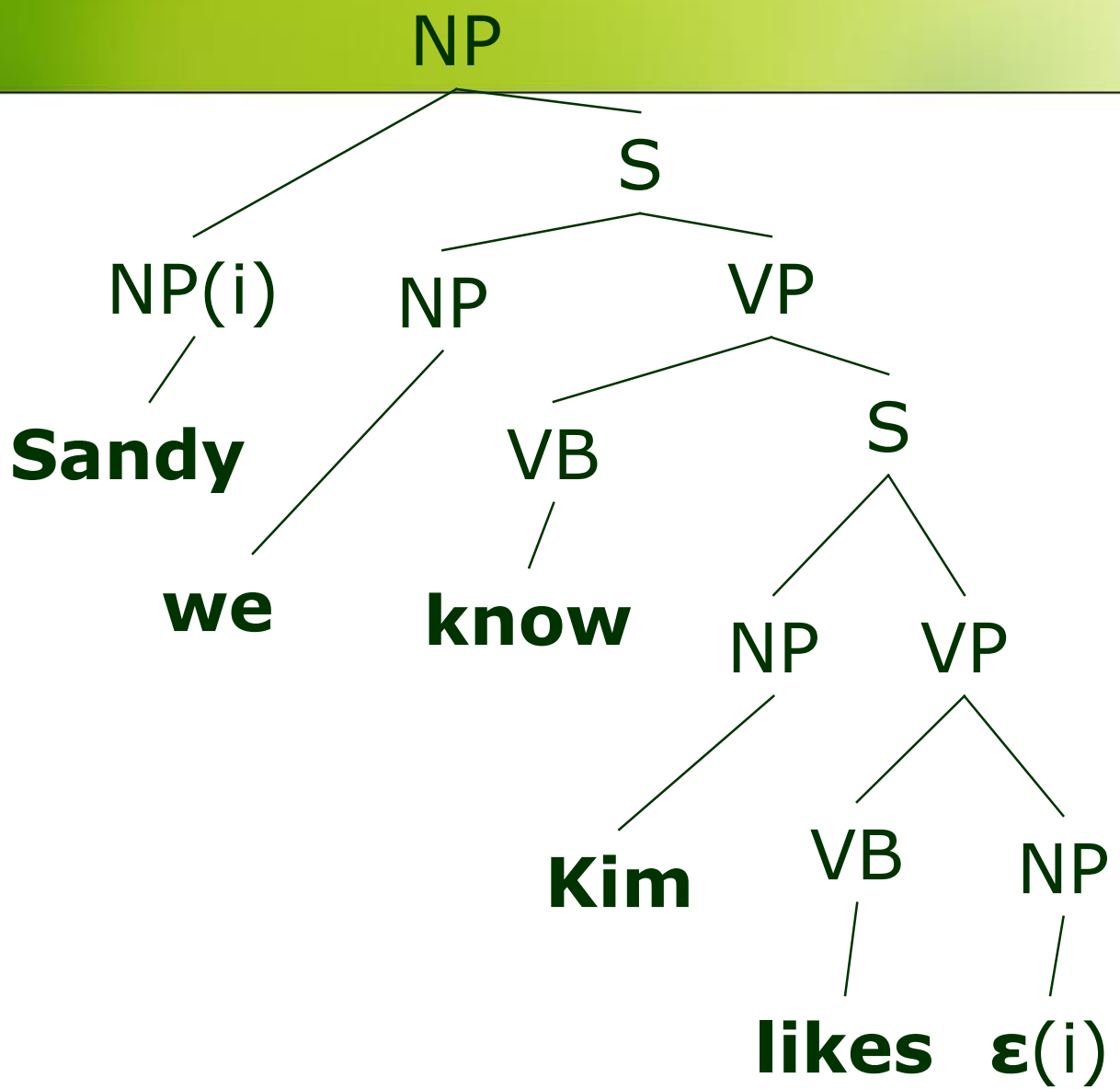




接合



Non-local Dependencies



言語学的考察

- R. Frank ``Phrase Structure Composition and Syntactic Dependencies'' MIT Press 2002

Extensions

- LTAG (Lexicalized Tree Adjoining Grammar)
  - 全ての基本木は必ず一つ以上のアンカーをもつ
- FB-LTAG (Feature-Based LTAG)
  - 構文木ノードに素性構造を追加したもの
  - 数や人称の制約を容易にいれられる
  - XTAG (Doran+1994, 1997)が有名

計算量

- TAGの計算量は一般に$O(n^6)$
  - nは文の単語長
  - C.f. CFGは$O(n^3)$
- Tree Insertion Grammar (TIG)
  - 補助木の足が左端あるいは右端にあらわれることだけを要求
  - $O(n^3)$
- Tree Substitution Grammar (TSG)
  - Adjoiningを使わないTAG (Substitutionのみ使う)
  - 部分木で記述するCFG (CFG≒TSG)
  - わかりやすい、確率モデルと相性が良い
  - $O(n^3)$

依存文法

依存文法: 導入

- 日本語の場合
  - 語順が比較的自由
  - 格要素の省略が可能
- 日本語だけではなく比較的自由な語順の言語は多い（語順が入れ替わる現象はスクランブリングと呼ばれる）
  - チェコ語（依存文法の研究で有名）
  - ドイツ語（句構造のスクランブリングの研究で有名）
  - …

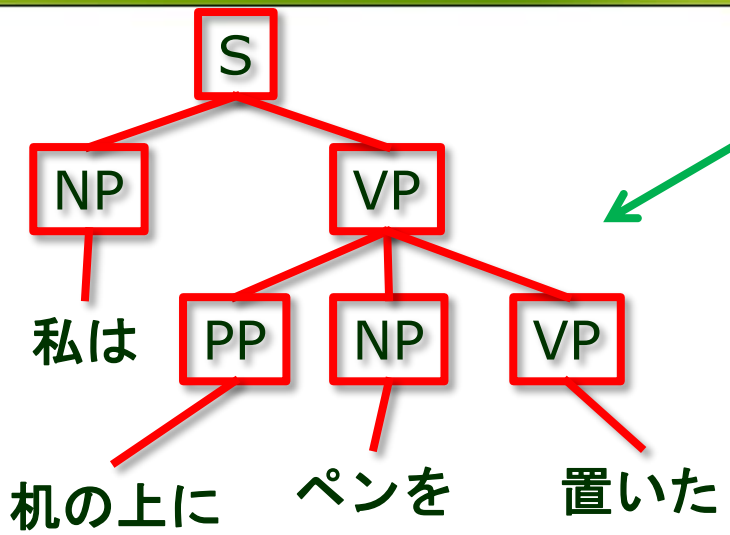
依存文法: 導入

- 英語の例
  - I put a pen on the table (ok)
  - *A pen put I on the table (NG)
  - * I put a pen (NG)
- 日本語の例
  - 私は机の上にペンを置いた。(ok)
  - ペンを私は机の上に置いた。(ok)
  - 私はペンを置いた。(ok)

句構造と依存文法

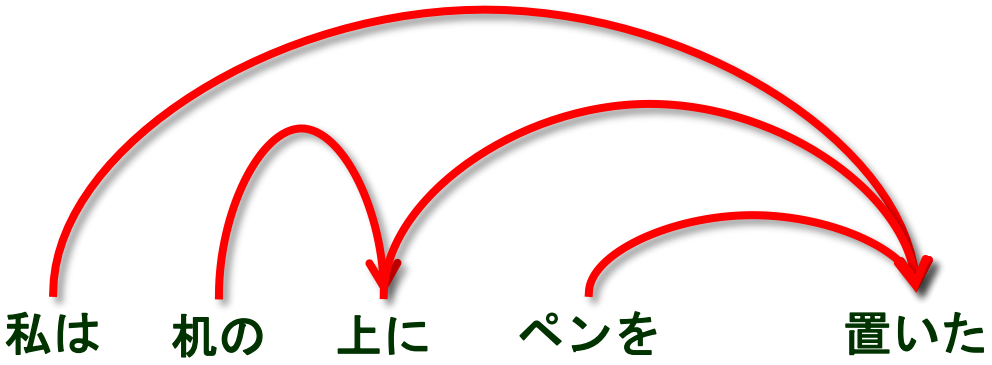
- 文法を大きくわけると二種類ある
  - 句構造文法
    - CFG
    - TAG
    - CCG
    - HPSG
    - LFG
  - 依存文法 (Hays 1964; Robinson 1970)
    - 特別な句構造を持たず、単語間の依存関係を記述する形態の文法

句構造と依存構造



句構造

依存構造



依存構造の文法

- 二つの依存可能な「係り元」「係り先」を記述
  - 簡単な日本語文法の例

係り元	係り先
名詞＋格助詞「が」	ガ格を持つ動詞
名詞＋格助詞「を」	ヲ格を持つ動詞
動詞　連用形	動詞、形容詞、形容動詞
動詞　連体形	名詞
動詞＋述語接続助詞	動詞、形容詞、形容動詞
副詞	動詞、形容詞、形容動詞
連体詞	名詞

依存文法の長所

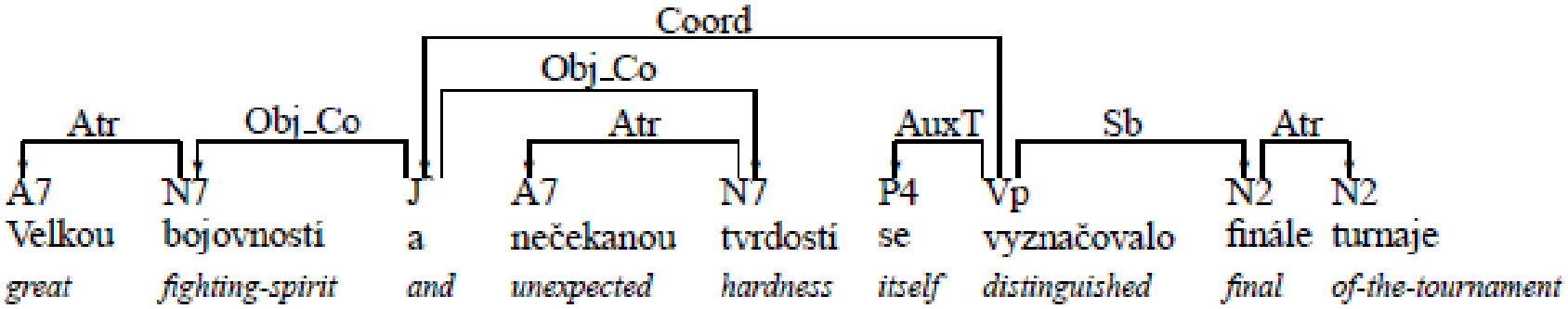
- 係り元と係り先の関係は個々に調べられるため、省略や語順に関係なく解析可能
- 格解析も同時解析可能
- 句構造解析でも語彙化の流れがあるが、依存文法はもともと単語を中心に考えられた文法である

機械学習による解析

- 単語と単語の依存構造を機械学習で判定
  - 陽に文法をもたない
  - 動的に生成されるデータ構造に強く依存しないためコンピュータ上で処理しやすい⇔句構造解析
  - エントロピー最大化法による日本語係り受け解析 (内元1998)
  - Support Vector Machine (SVM)による日本語係り受け解析 (工藤+2000)
- ツリーバンク（正解データの集合）の出現
  - 正解データがあれば手でルールを書くより機械学習の素性（特徴ベクトル）を用意するほうが簡単でかつ高性能

ツリーバンクの例

- プラハ依存構造ツリーバンク

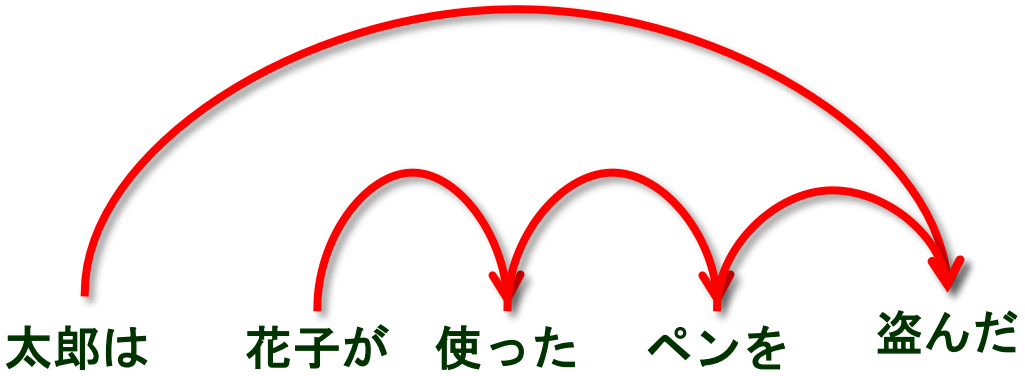


("The final of the tournament was distinguished by great fighting spirit and unexpected hardness")

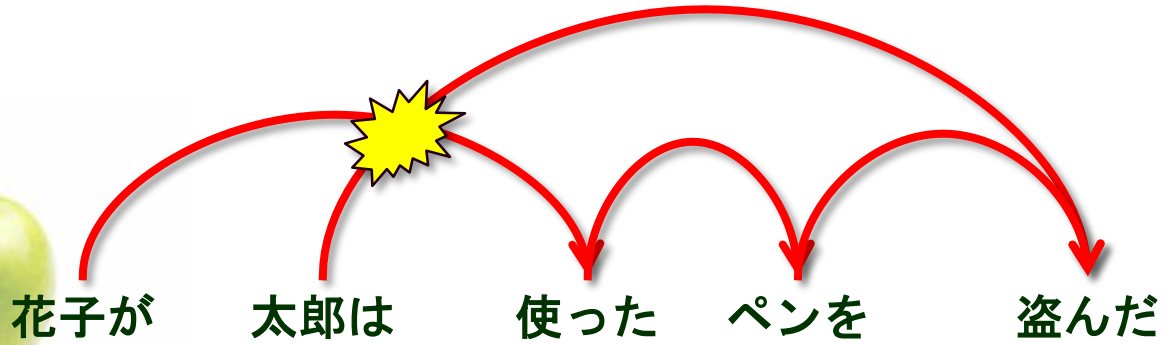
Figure 1: Dependency graph for a Czech sentence from the Prague Dependency Treebank

Projective or Non-projective

- 非交差条件



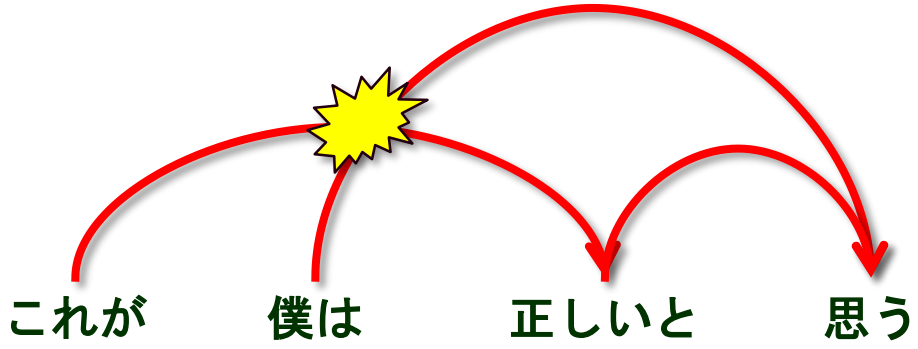
Good!



Bad!

Projective or Non-projective

- 非交差条件の例外



Projective or Non-projective

- Projective dependency parsing: 係り受けが交差することを想定しない解析
  - 句構造解析と同様の動的計画法が使えるため効率的 c.f. Eisner Algorithm O(n^3) (Eisner 1996)
- Non-Projective dependency parsing: 係り受けが交差することを想定した解析
  - 単語対の全ての組み合わせに対し解析するため非効率的（だと思われていた）
  - 単語＝頂点、係り受けの重み＝枝の重みとみなすと、maximum spanning treeの問題に帰着可 (McDonald+2005)
    - Chu-Liu-Edmond algorithm O(n^2) (Chu and Liu 1965; Edmond 1967; Tarjan 1977)

まとめ

- TAG
- 依存文法

- 資料

  http://aiweb.cs.ehime-u.ac.jp/~ninomiya/ai2/