人工知能特論II　第13回

二宮　崇

今日の講義の予定

- CFGの構文解析
  - CKYアルゴリズム
- PCFGの構文解析
  - CKYアルゴリズム+ビタビアルゴリズム
- 教科書
  - 北研二(著) 辻井潤一(編) 言語と計算4 確率的言語モデル 東大出版会
  - C. D. Manning & Hinrich Schütze “FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING” MIT Press, 1999
  - D. Jurafsky, J. H. Martin, A. Kehler, K.V. Linden & N. Ward “**Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**” Prentice Hall Series in Artificial Intelligence, 2000

CFG構文解析

CFGの構文解析

- ある文sが与えられた時、文法Gによって導出できる全ての構文木を導出する構文解析
- 何のために？
  - PCFG構文解析の基礎
  - 構文解析後に、確率計算を行って、最も良い構文木を選択する
  - パラメータ推定の際に構文木の候補集合が必要（学習方法によっては必要ない）

CFG構文解析のアルゴリズム

- トップダウン型
  - アーリー法 (earley parsing algorithm)
- ボトムアップ型
  - CKY法 (CKY parsing algorithm, CYK法ともいう)
  - チャート法 (chart parsing algorithm)
  - 左隅解析法 (left-corner parsing algorithm)
- 一般化LR法 (generalized LR parsing)

CKY法

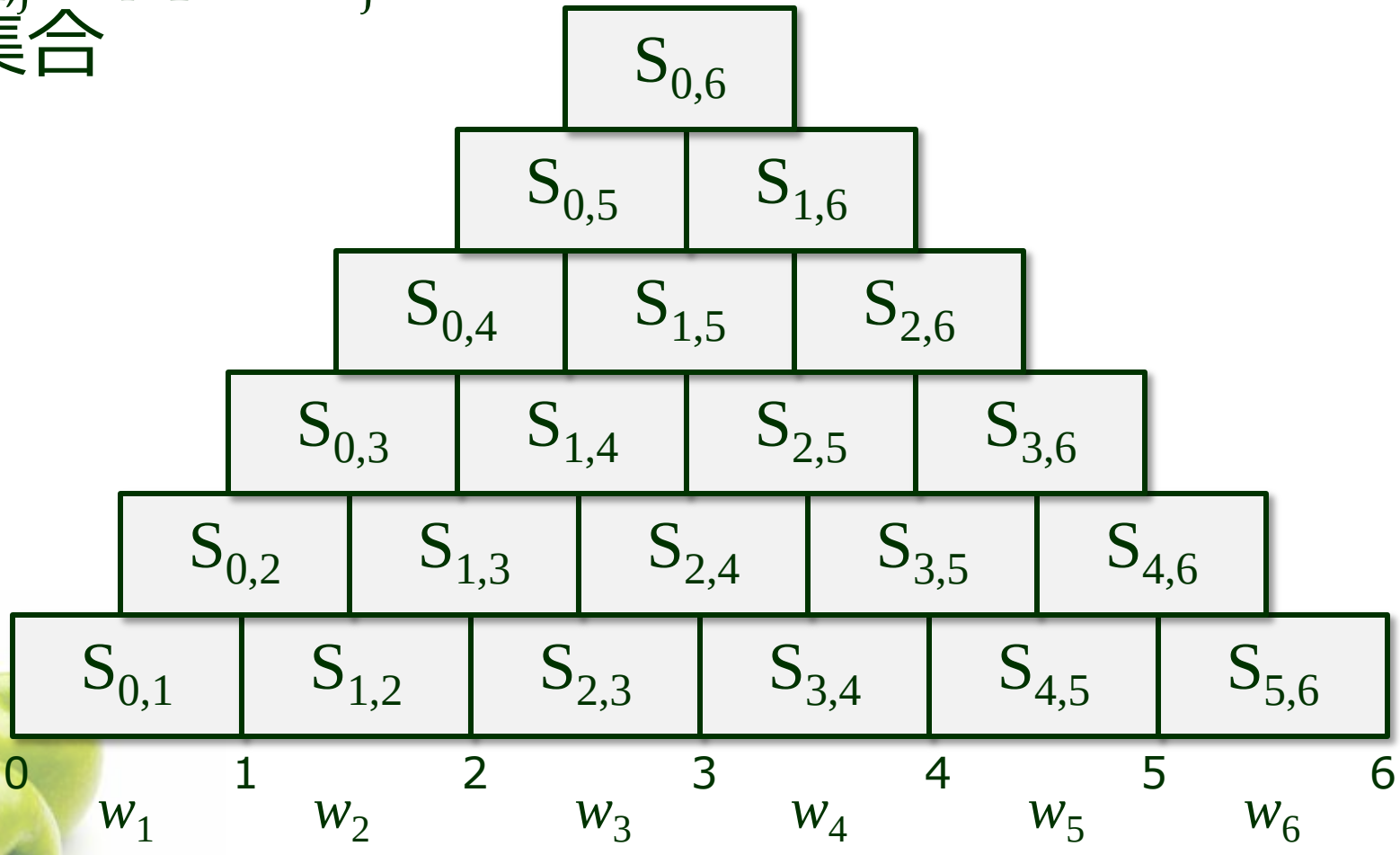
- Cocke, Kasami, Youngerにより提案され、それぞれの頭文字をとって、CKYもしくはCYK構文解析アルゴリズムと呼ばれる
- 多くのパーザーで用いられている
  - 簡単
  - 効率が良い
  - デコーディングと相性が良い
- 文法規則はバイナリルールかユーナリールールのみ
  - バイナリールール: 書換規則の右側の要素が二つしかないルール
  - ユーナリールール: 書換規則の右側の要素が一つしかないルール
  - CFGならチョムスキー標準形に変形
  - HPSG、CCGではバイナリールールを想定しているので特に問題は無い

準備: 書換規則と位置

- 書換規則は次の3つを想定
  - A → B C (バイナリールール)
  - A → B (ユーナリールール)
  - A → w (辞書ルール)
- 位置
  - 文 $w_1,w_2,\ldots,w_n$ が与えられた時、
    - 単語w_iの位置: <i-1, i>
    - 句 $w_i,\ldots,w_j$の位置: <i-1, j>

準備: CKYテーブル（チャート）

- $S_{i,j}$: w_{i+1},…, w_jに対応する句の非終端記号の集合

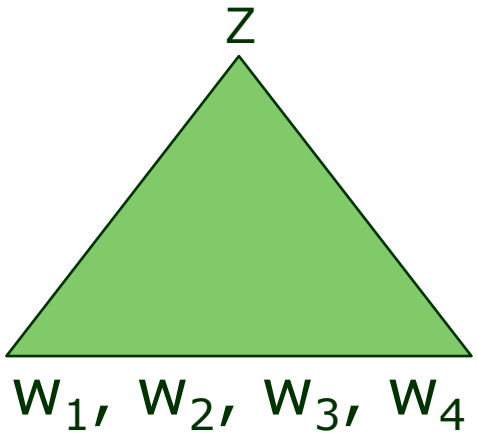


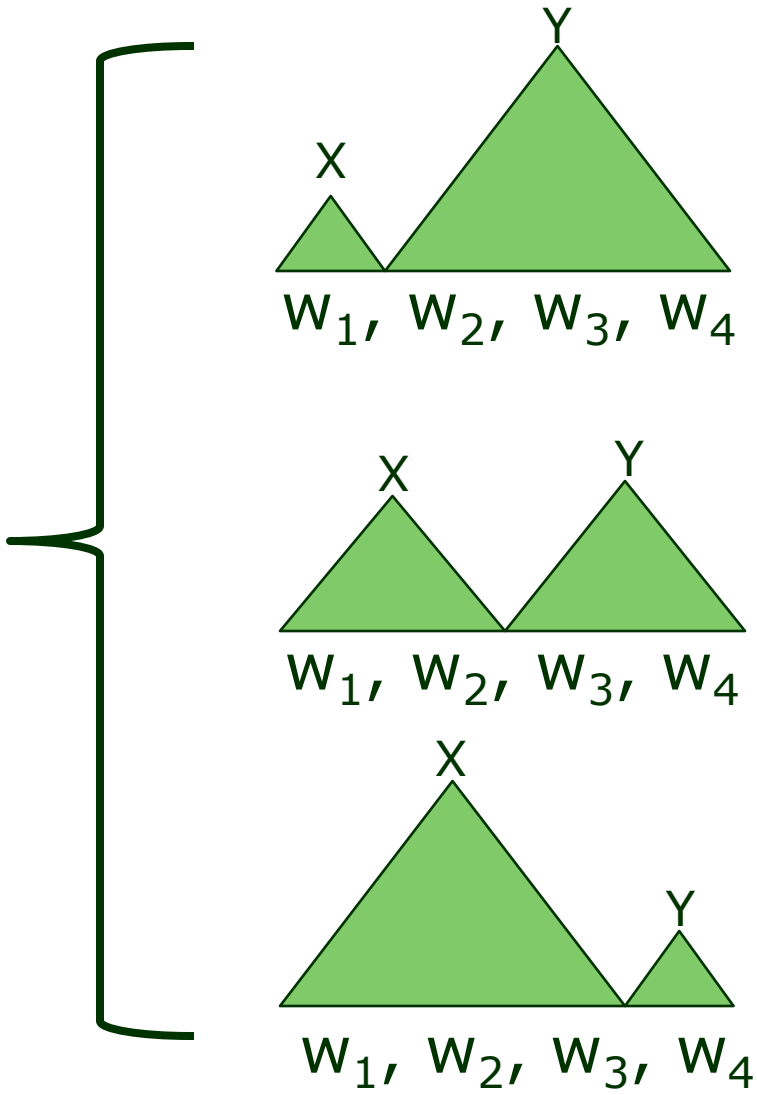
CKY法: 基本的なアイデア

- 目的: $S_{0, n}$を計算
- $S_{i,j}$は次のSから計算できる
  - $S_{i, i+1}$と$S_{i+1, j}$
  - $S_{i, i+2}$と$S_{i+2, j}$
  - ....
  - $S_{i, j-1}$と$S_{j-1, j}$

CKY法: 基本的なアイデア

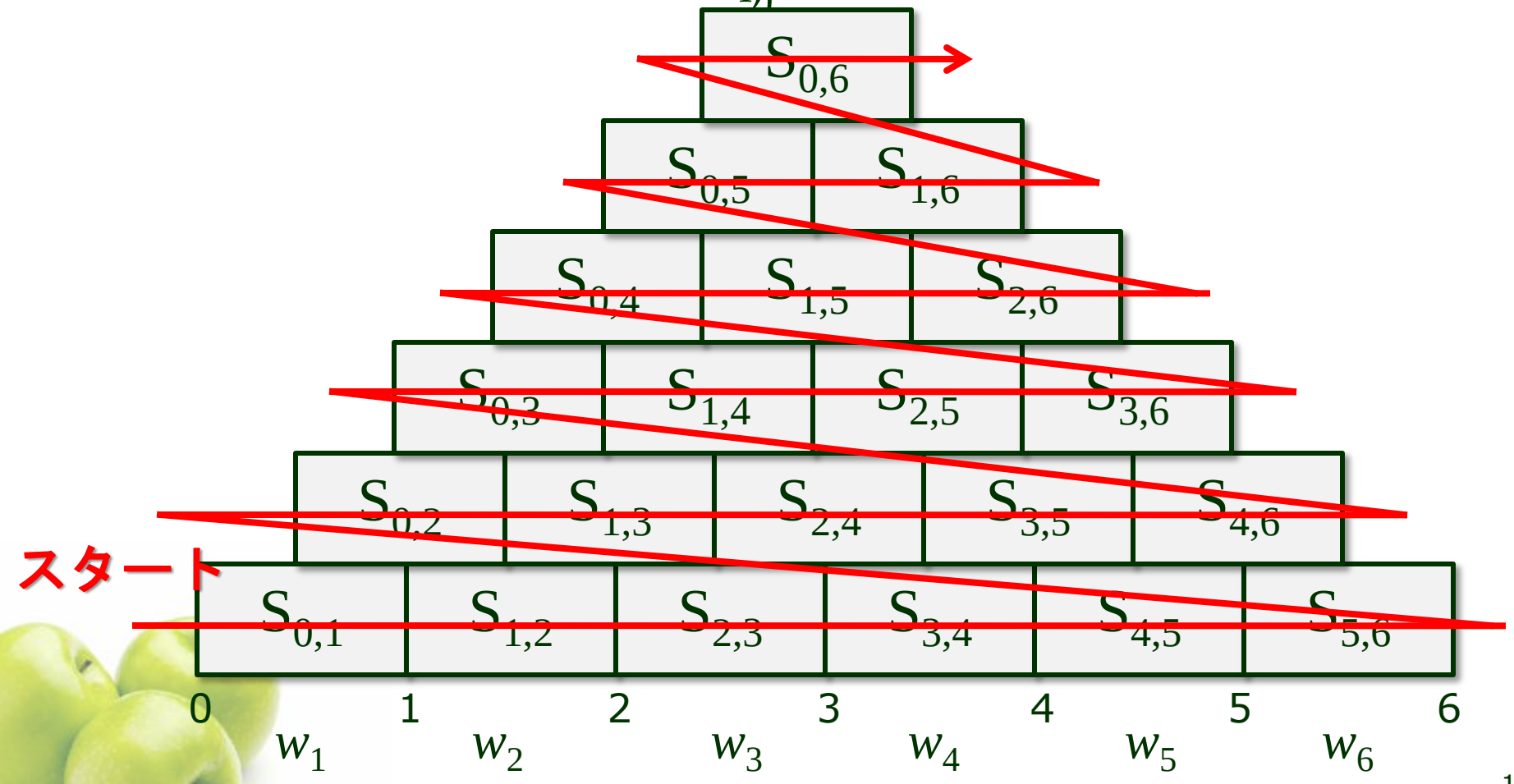
- $Z \rightarrow X\ Y$





CKY法

- 矢印の順で全ての$S_{i,j}$が求まる



ルール適用と$S_{i,j}$の求め方

- $G(X, Y) = \{Z \mid \exists p \in P.p=(Z \rightarrow X\ Y)\}$
  - $X\ Y$に対する全ての親を返す関数
    - X, Y: 非終端記号
    - P: 書換規則の集合
- $S_{i,j}$を求めるアルゴリズム

for $k = i+1$ to $j-1$

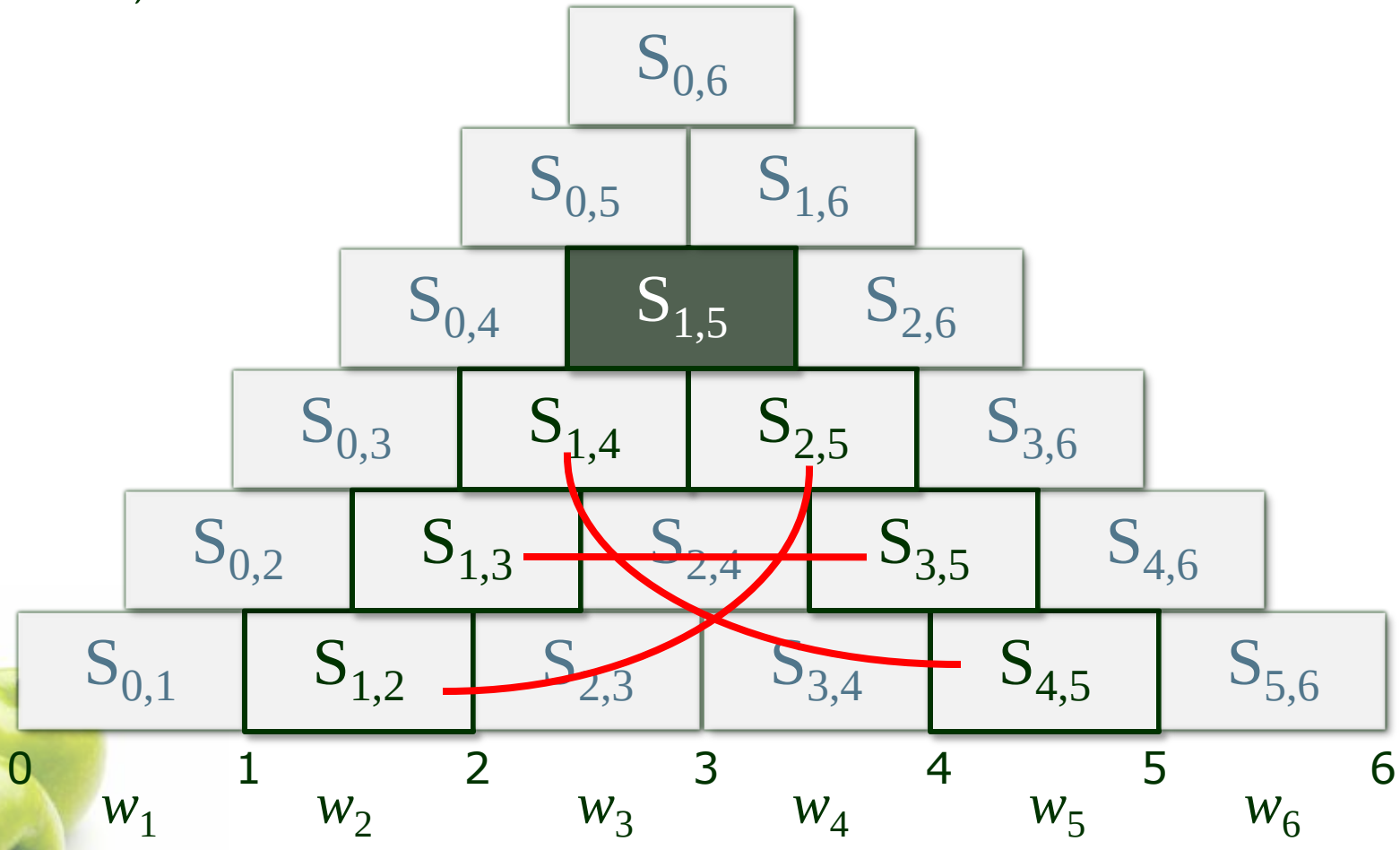
  forall $X \in S_{i,k}$

    forall $Y \in S_{k,j}$

      $S_{i,j} := S_{i,j} \cup G(X, Y)$

CKY法: $S_{i,j}$

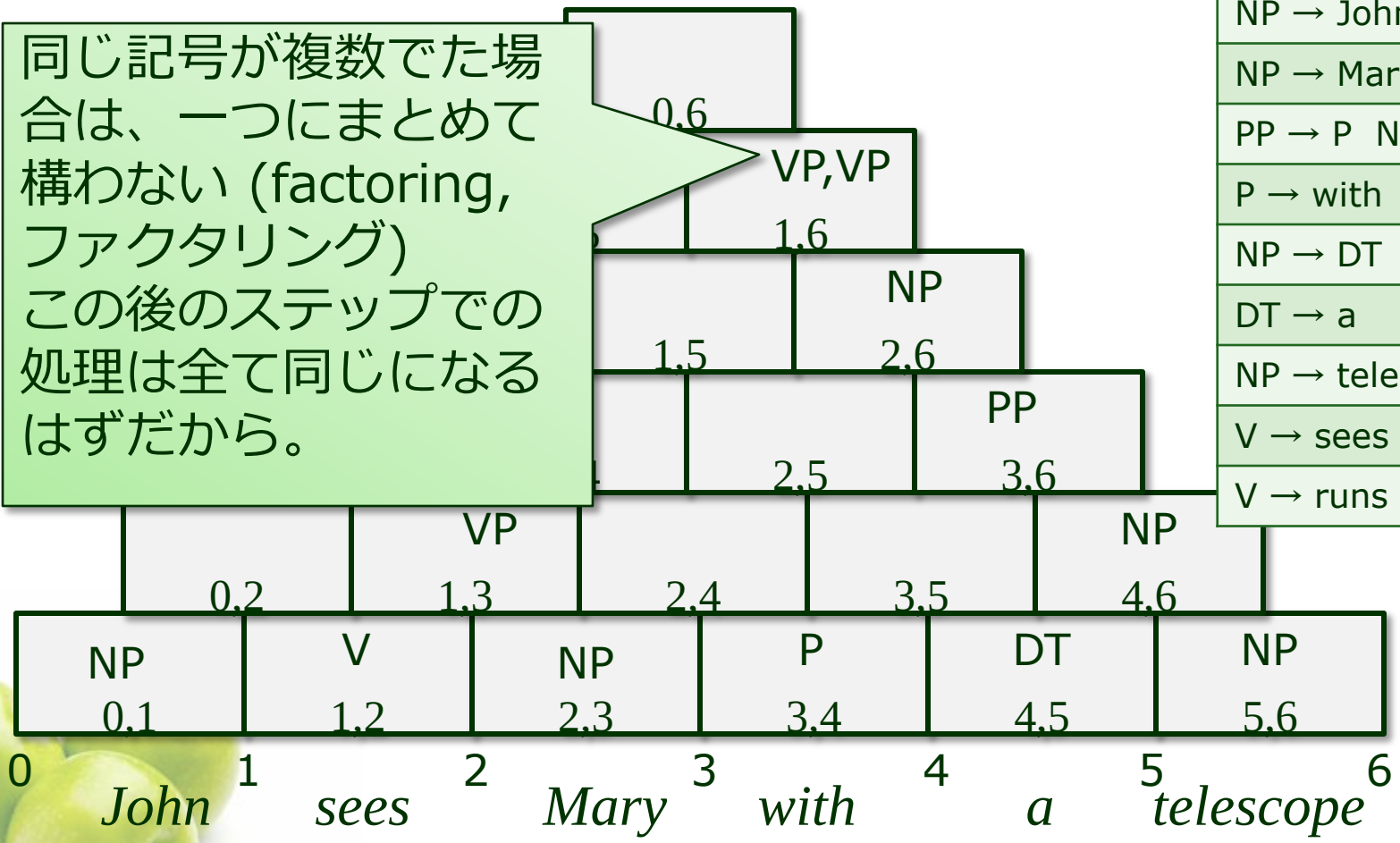
- 例: $S_{1,5}$に対しk=2,3,4



CKY法

- 例

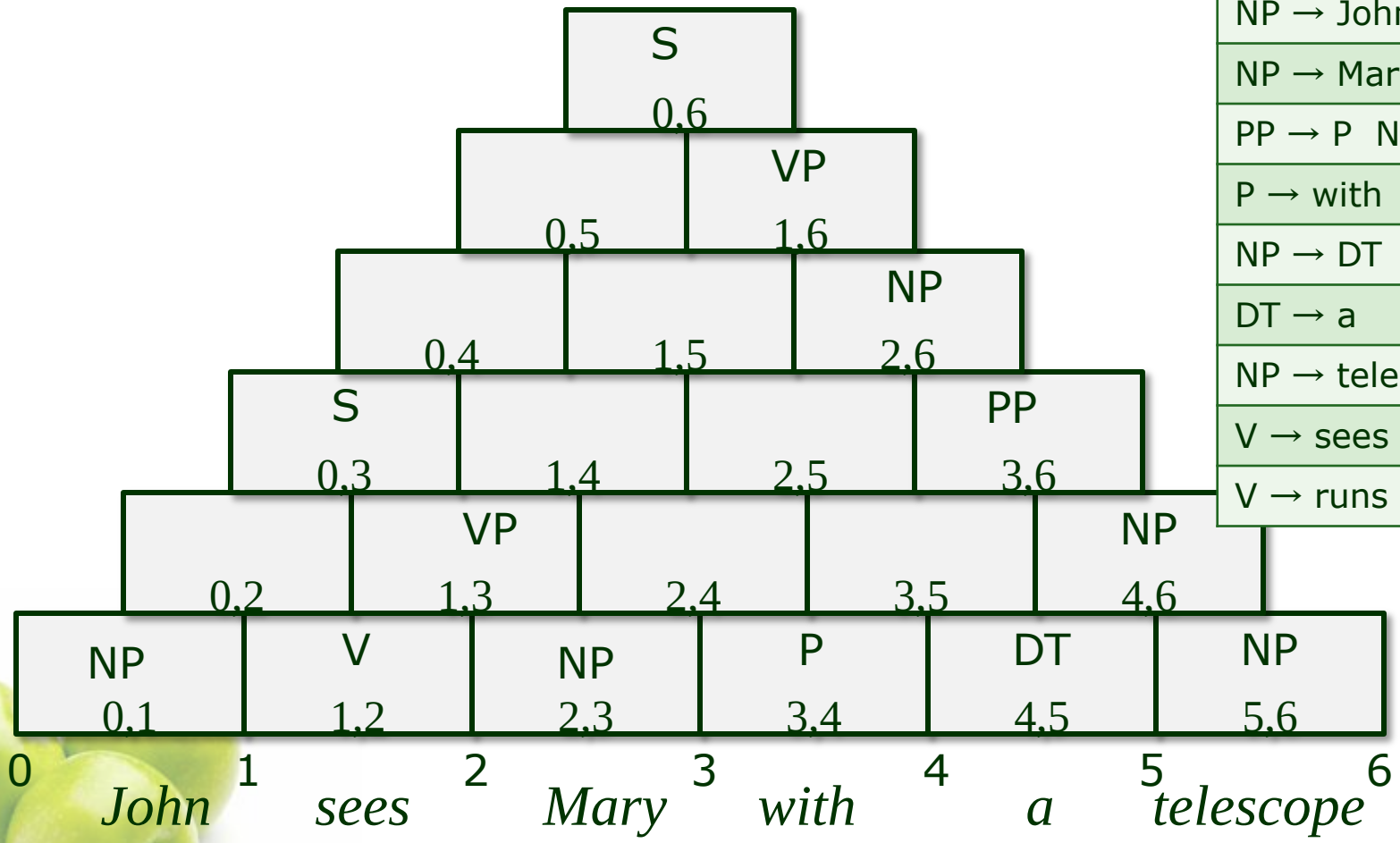
同じ記号が複数でた場合は、一つにまとめて構わない (factoring, ファクタリング)
この後のステップでの処理は全て同じになるはずだから。



文法
S → NP VP
VP → VP PP
VP → V NP
VP → V
NP → NP PP
NP → John
NP → Mary
PP → P NP
P → with
NP → DT NP
DT → a
NP → telescope
V → sees
V → runs

CKY法

- 例



文法
S → NP VP
VP → VP PP
VP → V NP
VP → V
NP → NP PP
NP → John
NP → Mary
PP → P NP
P → with
NP → DT NP
DT → a
NP → telescope
V → sees
V → runs

CKY法: アルゴリズム

for j = 1 to n

$S_{j-1,j} := L(w_j)$　## Lは単語wに対する非終端記号の集合を返す関数

for l = 2 to n

for i = 0 to n – l

j := i + l;

for k = i+1 to j - 1

forall $X \in S_{i,k}$

forall $Y \in S_{k,j}$

$Si,j := Si,j \cup G(X, Y)$

$S_{i,j} := S_{i,j} \cup U(S_{i,j})$ ## Uはユーナリールールを適用して得られる非終端記号集合

CKY法: 計算量

- 最悪時間計算量 (worst-case time complexity)
  - $O(n^3)$
    - nは文長
    - アルゴリズムより明らか
  - 非終端記号数を$|V_N|$とすると、$O(n^3|V_N|^2)$
  - ファクタリングのおかげで計算量が指数爆発していないということに注意！

PCFG

PCFG

- CFGの書換規則の適用確率をパラメータ化した文法
- 構文木の確率は、適用された書換規則のパラメータの積
- 各パラメータは $0.0 \leqq \theta \leqq 1.0$

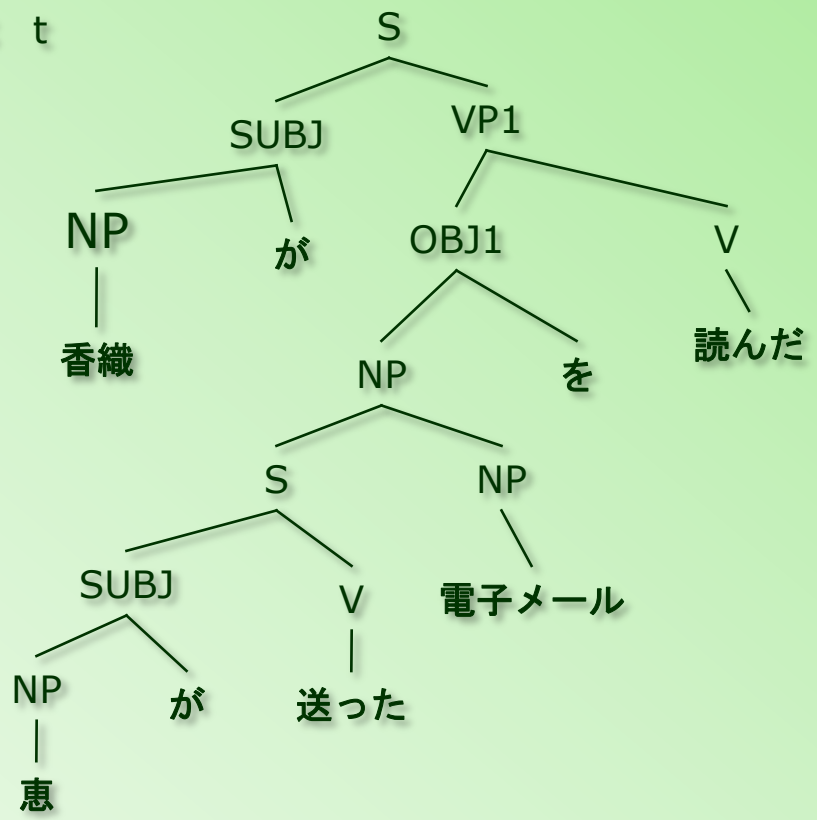
簡単なCFGの例	パラメータ
S → SUBJ　VP1	$\theta_{S \rightarrow SUBJ\ \ VP1}$
S → SUBJ　V	$\theta_{S \rightarrow SUBJ\ \ V}$
SUBJ → NP　が	$\theta_{SUBJ \rightarrow NP\ \ が}$
VP1 → OBJ1　V	$\theta_{VP1 \rightarrow OBJ1\ \ V}$
OBJ1 → NP　を	$\theta_{OBJ1 \rightarrow NP\ を}$
NP → S　NP	$\theta_{NP \rightarrow S\ \ NP}$
V → 送った	$\theta_{V \rightarrow 送った}$
V → 読んだ	$\theta_{V \rightarrow 読んだ}$
NP → 香織	$\theta_{NP \rightarrow 香織}$
NP → 恵	$\theta_{NP \rightarrow 恵}$
NP → 電子メール	$\theta_{NP \rightarrow 電子メール}$
NP → プレゼント	$\theta_{NP \rightarrow プレゼント}$
NP → 香織　NP1	$\theta_{NP \rightarrow 香織\ NP1}$
NP → 恵　NP1	$\theta_{NP \rightarrow 恵\ \ NP1}$
NP1 → と　NP	$\theta_{NP1 \rightarrow と\ \ NP}$

構文木の確率

簡単なCFGの例	パラメータ
S → SUBJ VP1	$\theta_{S \rightarrow SUBJ\ VP1}$
S → SUBJ V	$\theta_{S \rightarrow SUBJ\ V}$
SUBJ → NP が	$\theta_{SUBJ \rightarrow NP\ が}$
VP1 → OBJ1 V	$\theta_{VP1 \rightarrow OBJ1\ V}$
OBJ1 → NP を	$\theta_{OBJ1 \rightarrow NP\ を}$
NP → S NP	$\theta_{NP \rightarrow S\ NP}$
V → 送った	$\theta_{V \rightarrow 送った}$
V → 読んだ	$\theta_{V \rightarrow 読んだ}$
NP → 香織	$\theta_{NP \rightarrow 香織}$
NP → 恵	$\theta_{NP \rightarrow 恵}$
NP → 電子メール	$\theta_{NP \rightarrow 電子メール}$
NP → プレゼント	$\theta_{NP \rightarrow プレゼント}$
NP → 香織 NP1	$\theta_{NP \rightarrow 香織\ NP1}$
NP → 恵 NP1	$\theta_{NP \rightarrow 恵\ NP1}$
NP1 → と NP	$\theta_{NP1 \rightarrow と\ NP}$

文 s = “香織が恵が送った電子メールを読んだ”

構文木 t



$$P(t) = \theta_{S \rightarrow SUBJ\ VP1} \times \theta_{SUBJ \rightarrow NP\ が} \times \theta_{NP \rightarrow 香織} \times \theta_{VP1 \rightarrow OBJ1\ V} \times \theta_{OBJ1 \rightarrow NP\ を} \times \theta_{NP \rightarrow S\ NP} \times \theta_{S \rightarrow SUBJ\ V} \times \theta_{SUBJ \rightarrow NP\ が} \times \theta_{NP \rightarrow 恵} \times \theta_{V \rightarrow 送った} \times \theta_{NP \rightarrow 電子メール} \times \theta_{V \rightarrow 読んだ}$$

書換規則の制約

- CFG書換規則をA→ αと表したとき、(Aは非終端記号、αは非終端記号列)すべての非終端記号Aに対し、

$$\sum_{\alpha} \theta_{A \to \alpha} = 1.0$$

とする。

構文木tの確率

- C(r; t): CFG<V_N, V_T, P, σ>の書換規則$r \in P$が構文木t中で使われた回数

$$p(t) = \prod_{r \in P} \theta_r^{C(r;t)}$$

PCFGの構文解析
～最大確率の木を選ぶ～

- ある文sに対し、CFG<V_N, V_T, P, σ>を用いてsを導出できる全ての構文木集合を$T(s)$としたとき、

$$\tilde{t} = \arg\max_{t \in T(s)} p(t)$$

文の生成確率

- ある文sに対し、sを導出する全ての構文木集合を$T(s)$としたとき、

$$p(s) = \sum_{t \in T(s)} p(t) \qquad 0.0 \leq p(s) \leq 1.0$$

- PCFGによって生成されうる全ての構文木集合をTとしたとき、

$$\sum_{t \in T} p(t) \approx 1.0$$

最大確率の木を選ぶアルゴリズム

PCFG構文解析

最大確率の木を選ぶアルゴリズム

- ビタビアルゴリズム (viterbi algorithm)
  - CFG構文解析を行うと同時に構文木の確率を計算する手法
  - CKYテーブルには、非終端記号と確率値のペアを格納
  - ファクタリングの際には最大確率のペアのみ残す
    - c.f. maxの代わりにsumを求めると、全ての構文木の確率の和(=文の生成確率)が求まる
  - 最適解
  - 効率は悪い

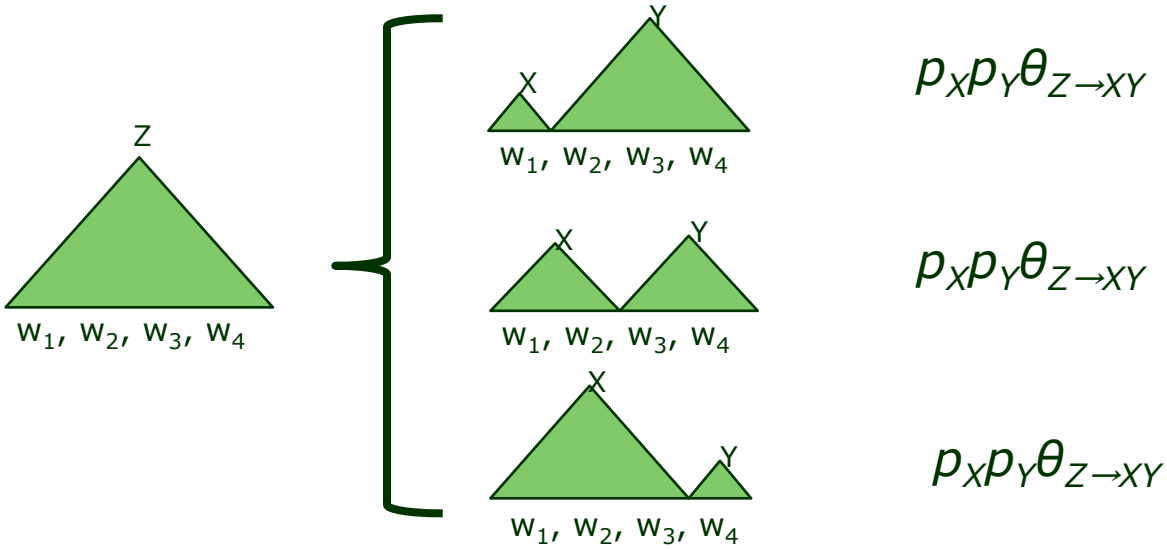
ビタビアルゴリズム:基本的なアイデア

- あるi, jに対し、

$$i < k < j, \langle X, p_X \rangle \in S_{i,k}, \langle Y, p_Y \rangle \in S_{k,j}, Z \to XY$$

を満たすk, Zが存在するなら、

$$\left\langle Z, \max_{k:i<k<j, \langle X, p_X \rangle \in S_{i,k}, \langle Y, p_Y \rangle \in S_{k,j}} p_X p_Y \theta_{Z \to XY} \right\rangle \in S_{i,j}$$



ビタビアルゴリズム

- Si,j: <X, p>の集合
  - X: 非終端記号
  - p: 部分木の確率
- $S_{i,j}$の求め方 (CKY法の場合)

for $k = i+1$ to $j-1$

forall $<X, p_X> \in S_{i,k}$

forall $<Y, p_Y> \in S_{k,j}$

forall $Z \in G(X, Y)$

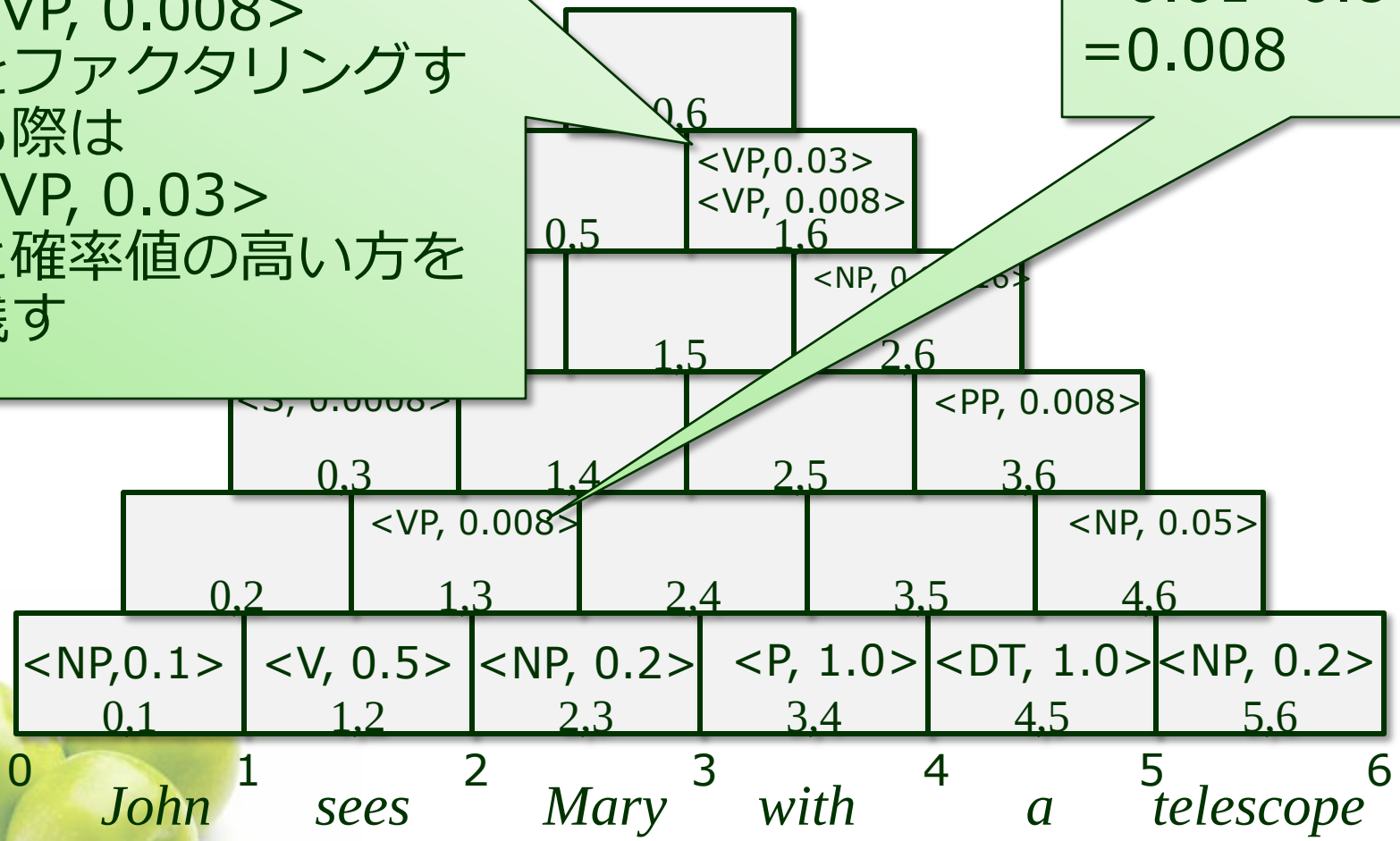
$S_{i,j} := S_{i,j} \cup <Z, p_X \times p_Y \times \theta_{Z \to X\,Y}>$

ファクタリング(同じ非終端記号が出現した場合の畳込み)の際には確率の高い方を選ぶ

ビタビアルゴリズム

<VP, 0.03>
<VP, 0.008>
をファクタリングする際は
<VP, 0.03>
と確率値の高い方を残す

$0.5 \times 0.2 \times \theta_{VP \rightarrow V\ NP}$
$=0.01 \times 0.8$
$=0.008$

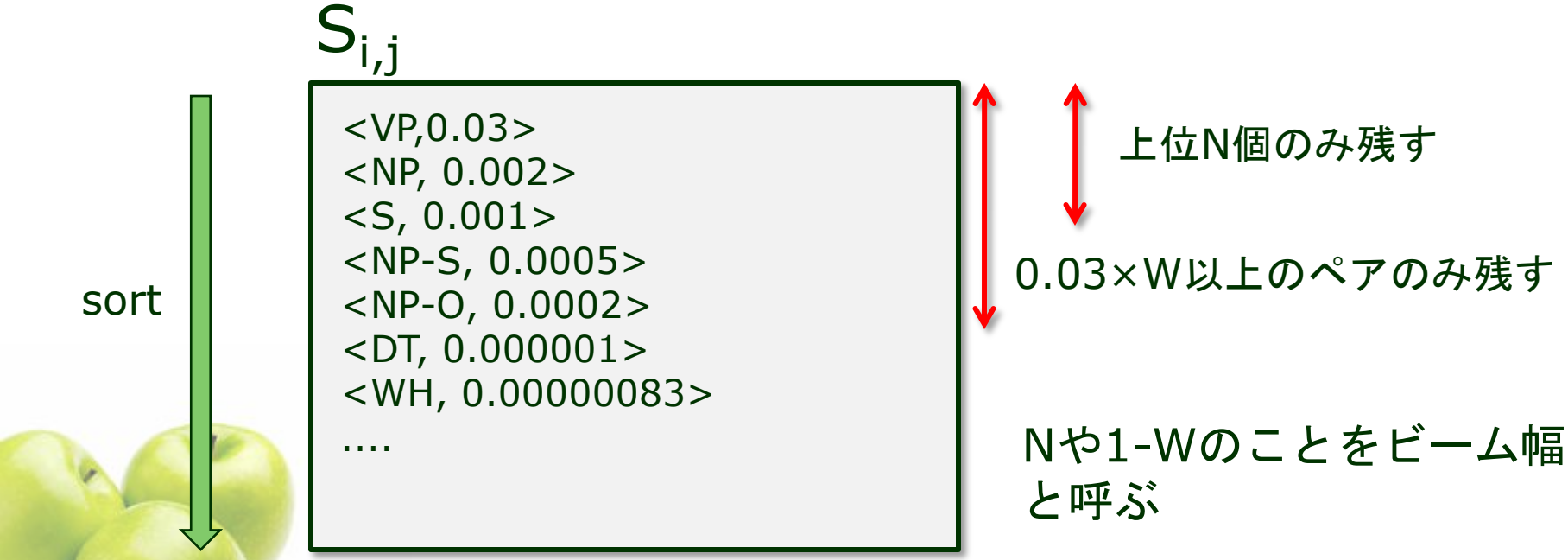


ビームサーチパージング (beam search parsing)

- ビタビアルゴリズムで解析する途中で、確率値の低い非終端記号を除去する
- 最適解は保障されない
- 効率は良い

ビームサーチ

- ２つの刈り方
  - 上位N個のみ残す
  - トップの確率×W以上の確率のみ残す



ビームサーチ

- N: 数による閾値
- W: 幅による閾値
- $S_{i,j}$の求め方 (CKY法の場合)

for $k = i+1$ to $j-1$
  forall $\langle X, p_X \rangle \in S_{i,k}$
    forall $\langle Y, p_Y \rangle \in S_{k,j}$
      forall $Z \in G(X, Y)$
        $S_{i,j} := S_{i,j} \cup \langle Z, p_X \times p_Y \times \theta_{Z \to X\ Y} \rangle$
sort $S_{i,j}$ according to its p ($S_{i,j}=\langle X_1, p_1 \rangle \ldots \langle X_M, p_M \rangle$とする)
remove $\langle X_{N+1}, p_{N+1} \rangle \ldots \langle X_M, p_M \rangle$
remove $\langle X, p \rangle \in S_{i,j}$ s.t. $p < p_1 \times W$

まとめ

- CFGの構文解析
  - CKYアルゴリズム
- PCFGの構文解析
  - CKYアルゴリズム+ビタビアルゴリズム
- 資料

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