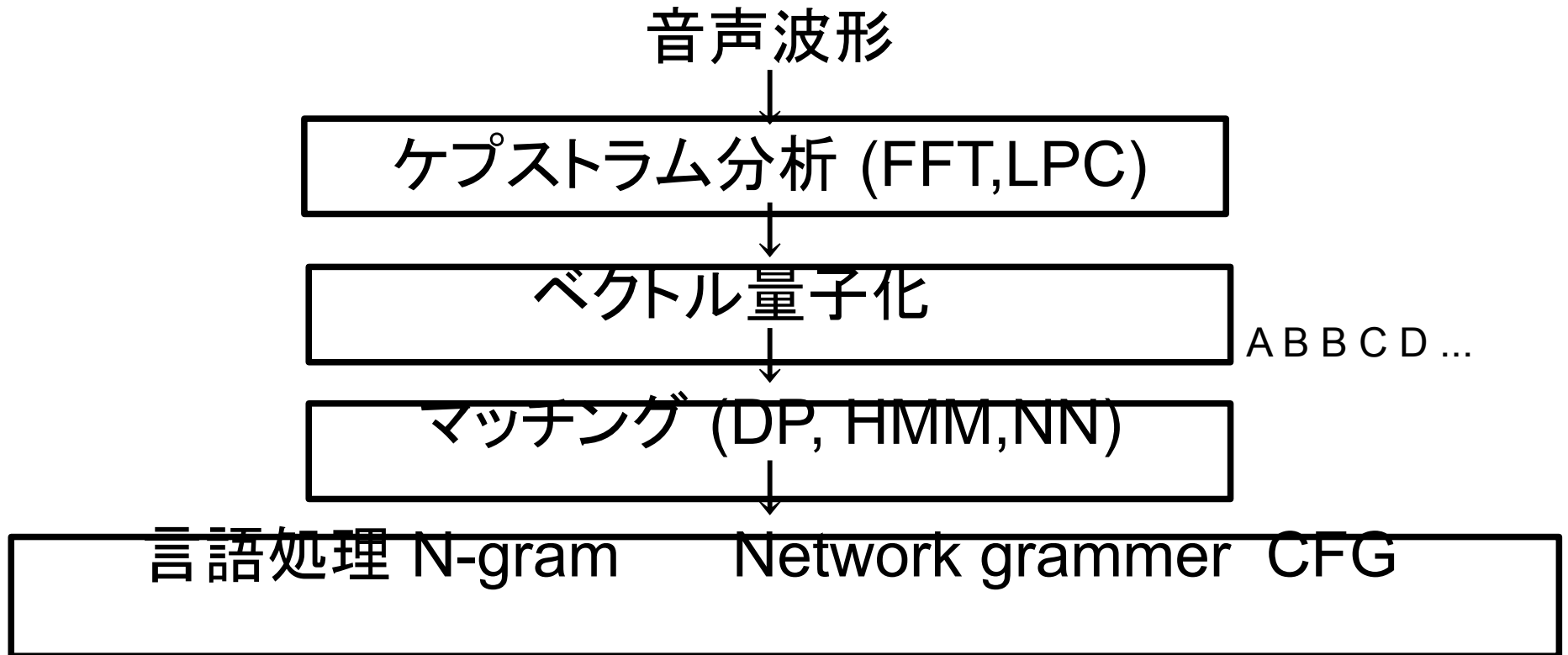


音声認識の方法



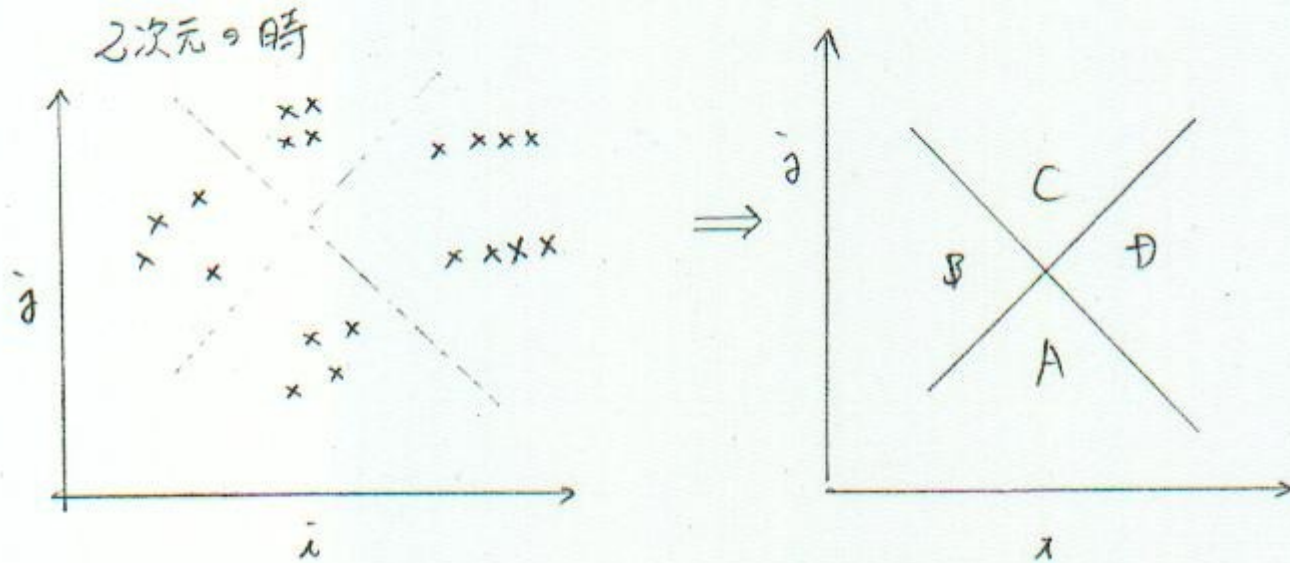
ベクトル量子化 (VQ)

アナログ量をベクトル量に変換



ケプストラム16次

256のVQへ変換



124 (100)

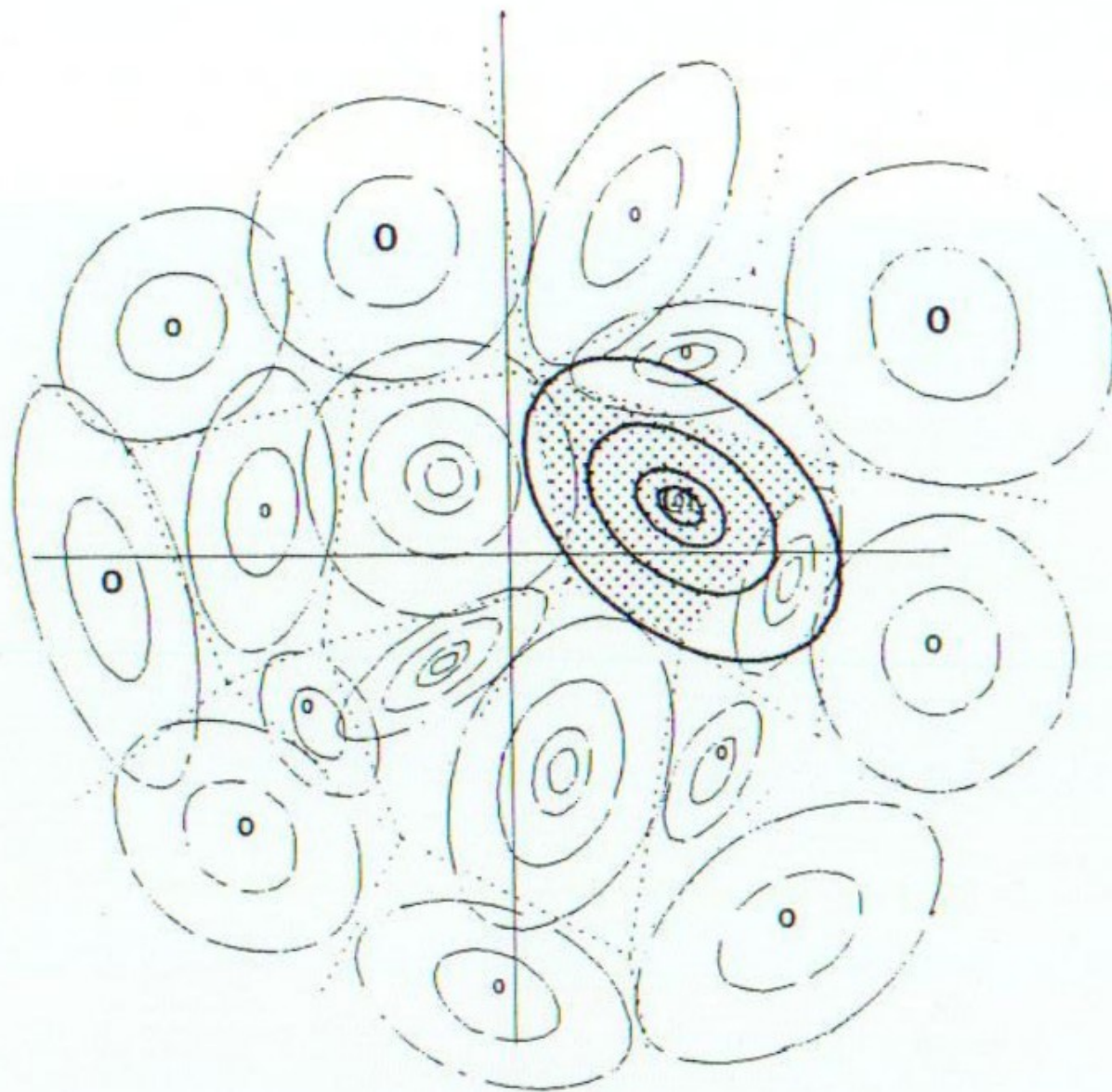


Figure 4.2.1. Partitioning of two-dimensional space with densities

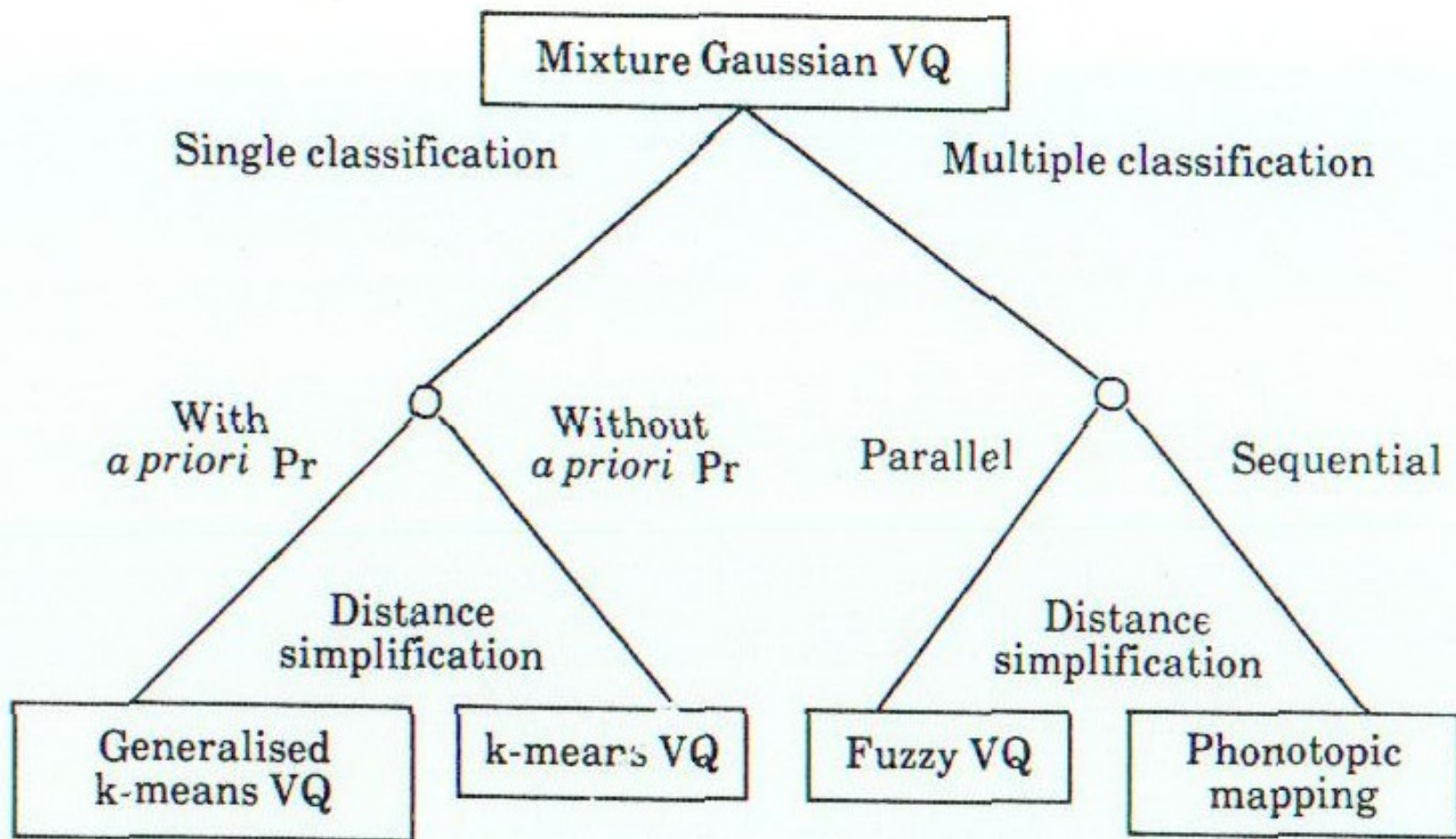
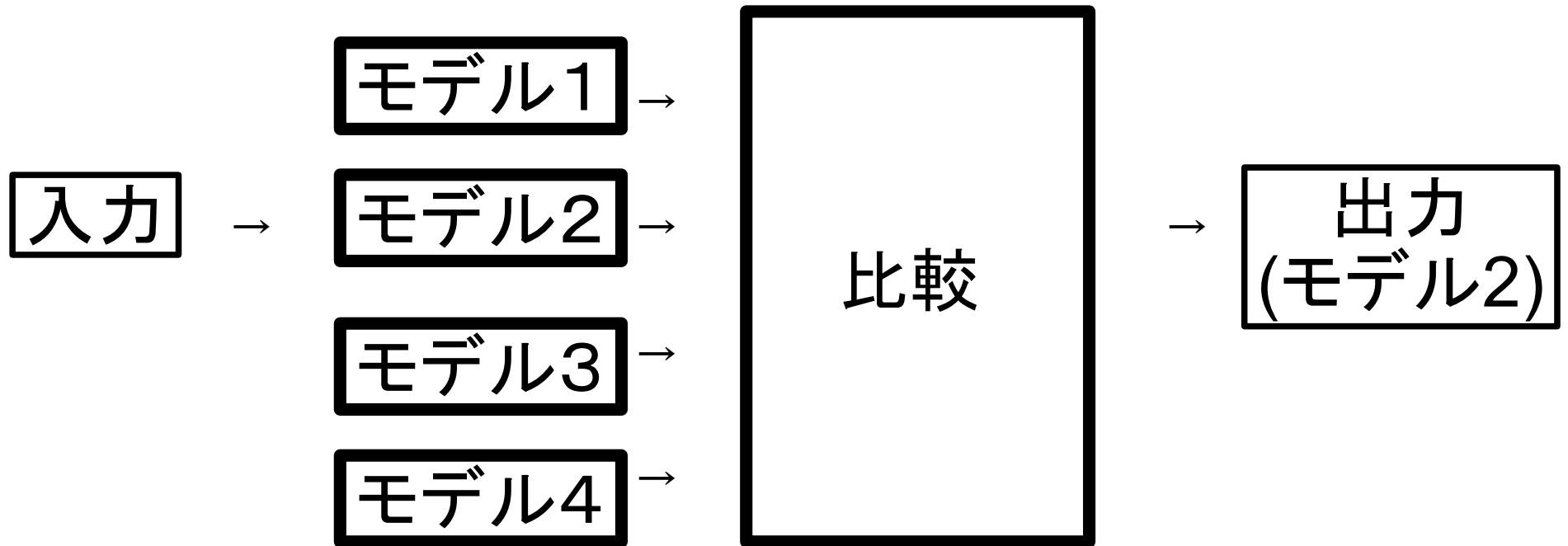


Figure 4.2.2. Relation of several VQ techniques.

計算機における認識の原理



すべてのモデルの尤度を計算

最も尤度の高いモデル: 認識結果

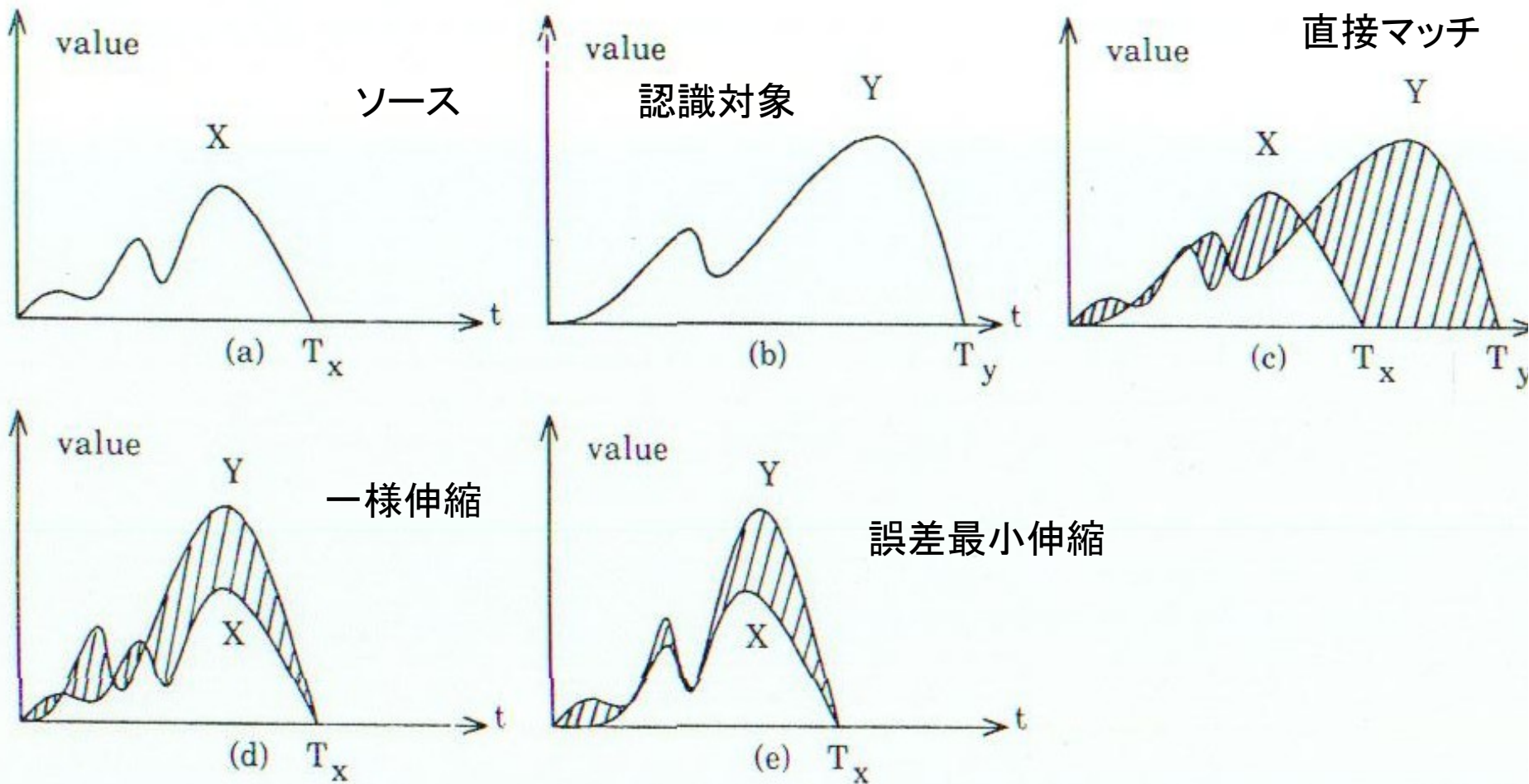


Figure 3.2.1. Acoustic patterns and their matching measure:
 (a)(b)acoustic pattern x and y, (c)matching without alignment,
 (d)linear matching and (e)non-linear matching.

音声認識におけるモデルの困難さ

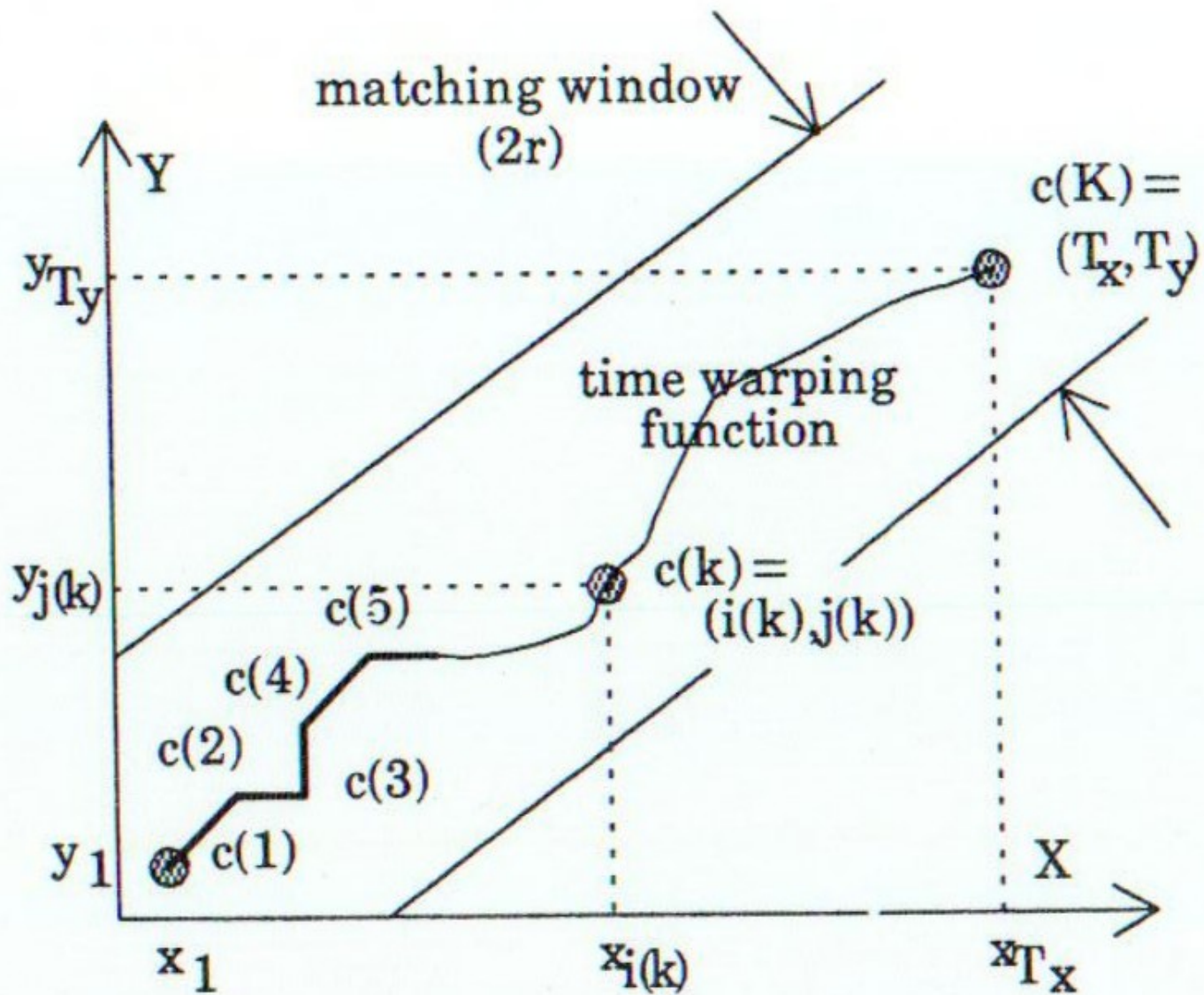
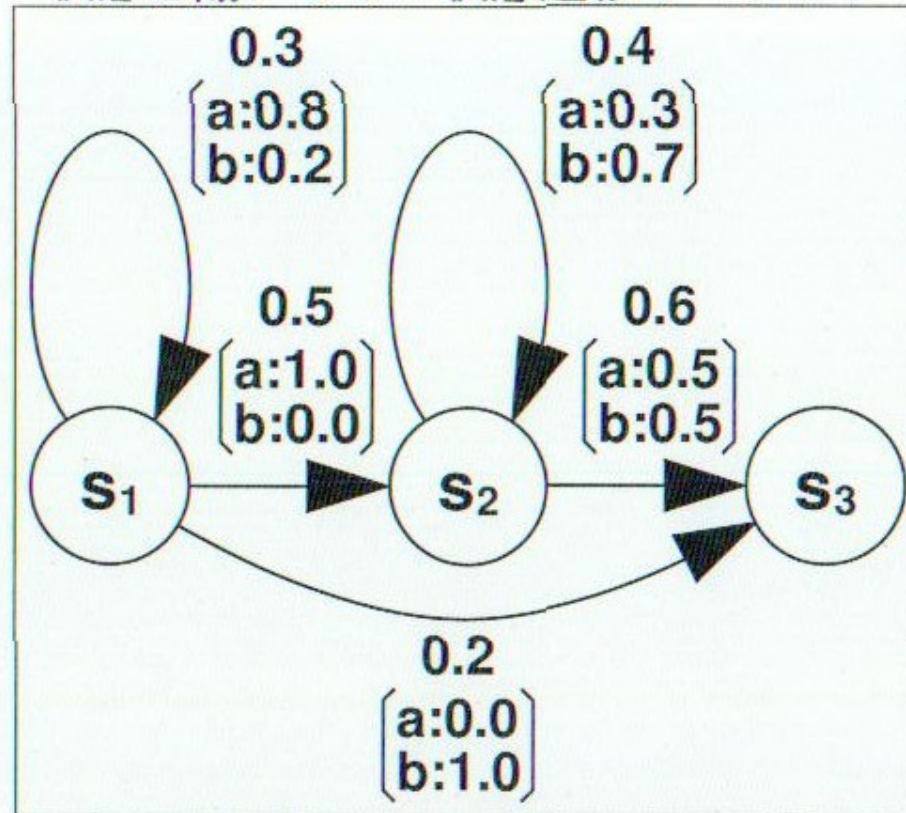


Figure 3.2.2. Non-linear matching in the DTW algorithm

HMM (Hidden Markov Model)

- 状態を陽にしない状態遷移オートマトン



3 状態 left-to-right HMM

S_1, S_2, S_3 : 状態

a, b: シンボル出力確率

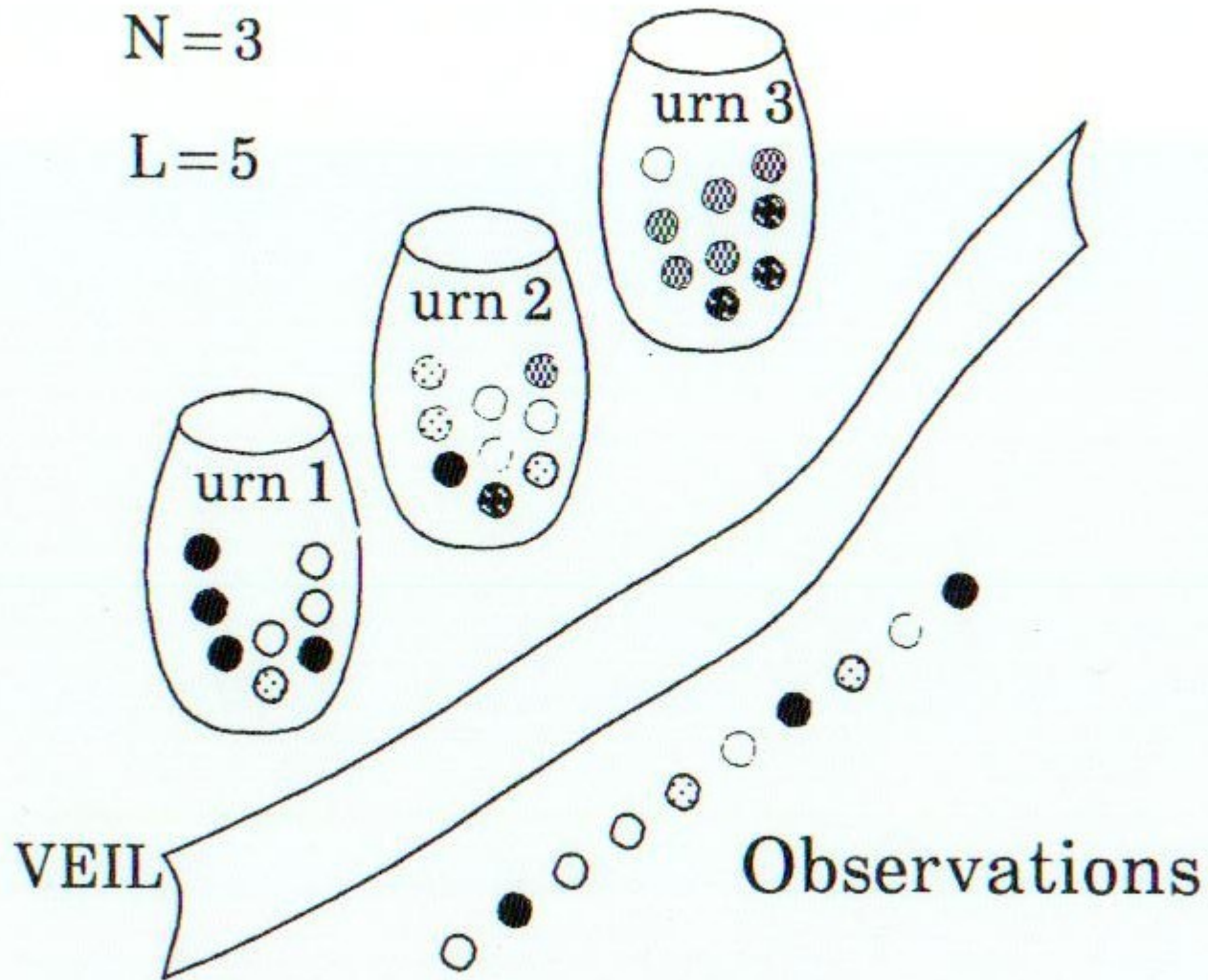


Figure 5.2.1. The urn and ball experiment.

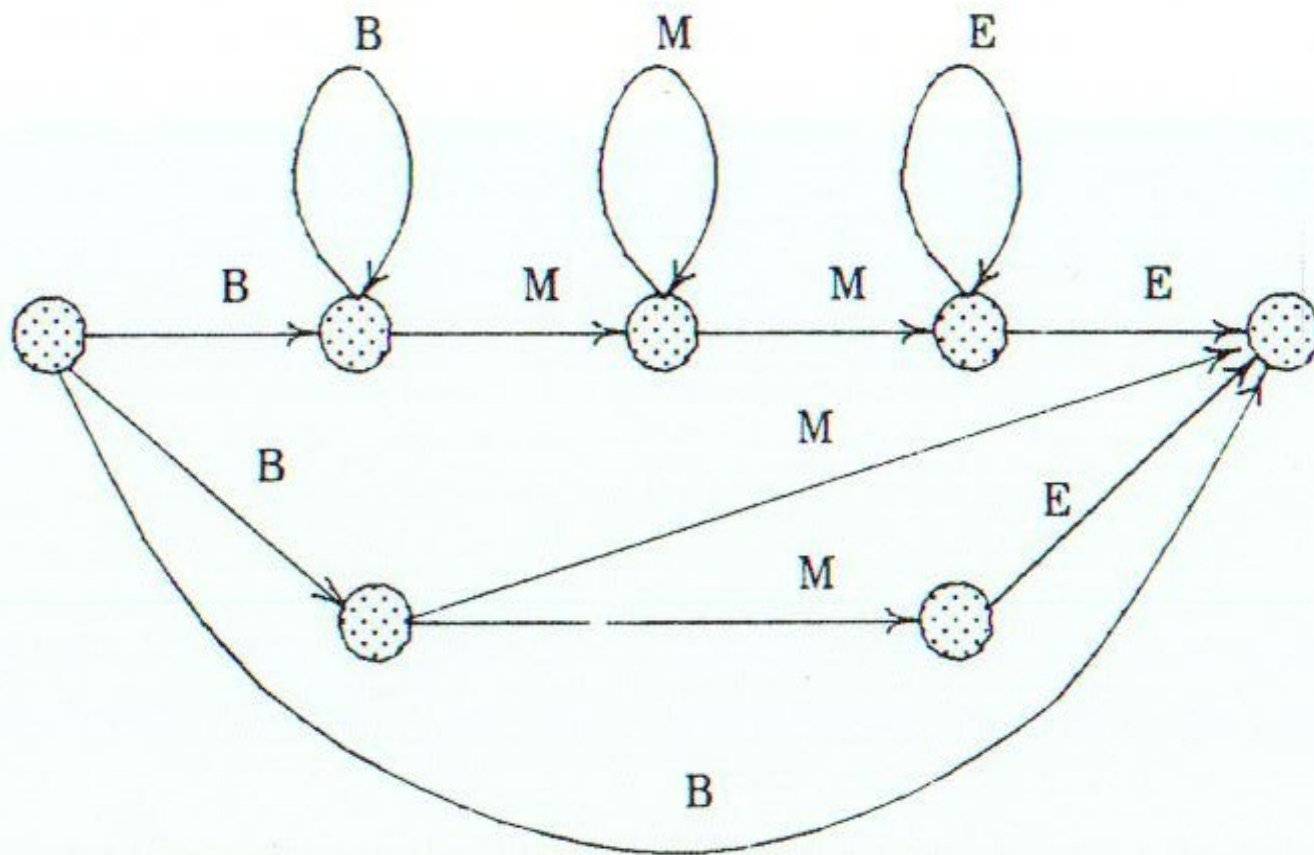


Figure 9.1.1. HMM used in SPHINX

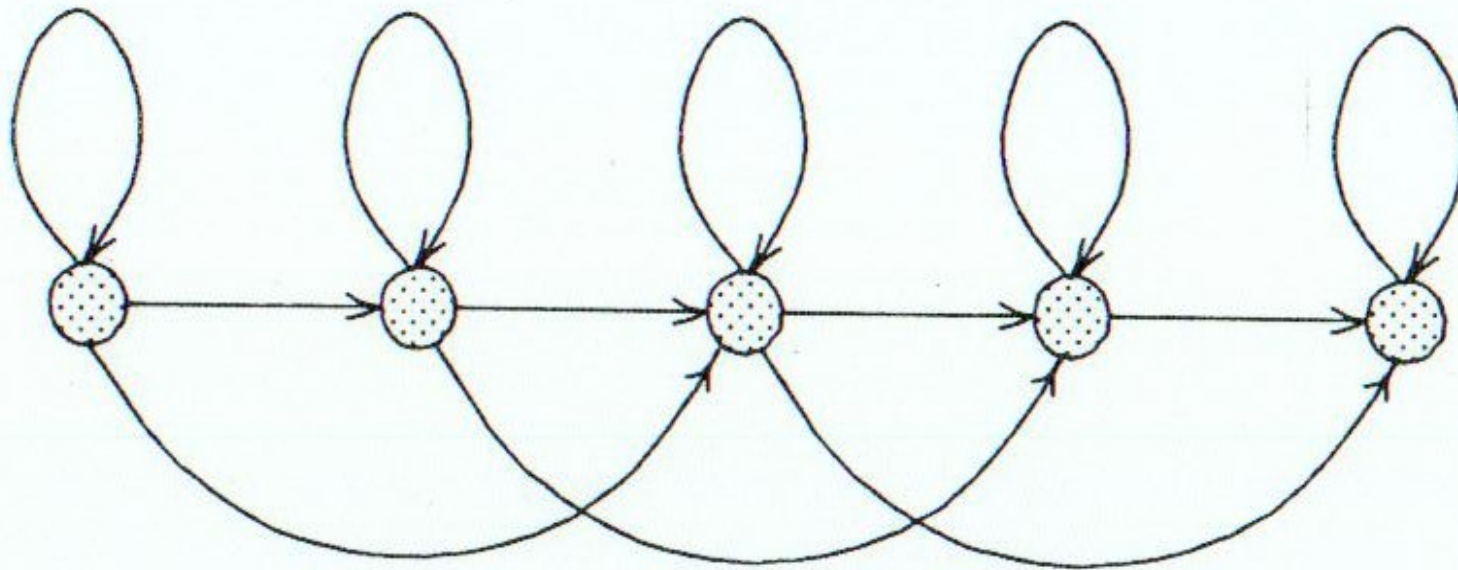


Figure 9.1.2. A left-to-right HMM

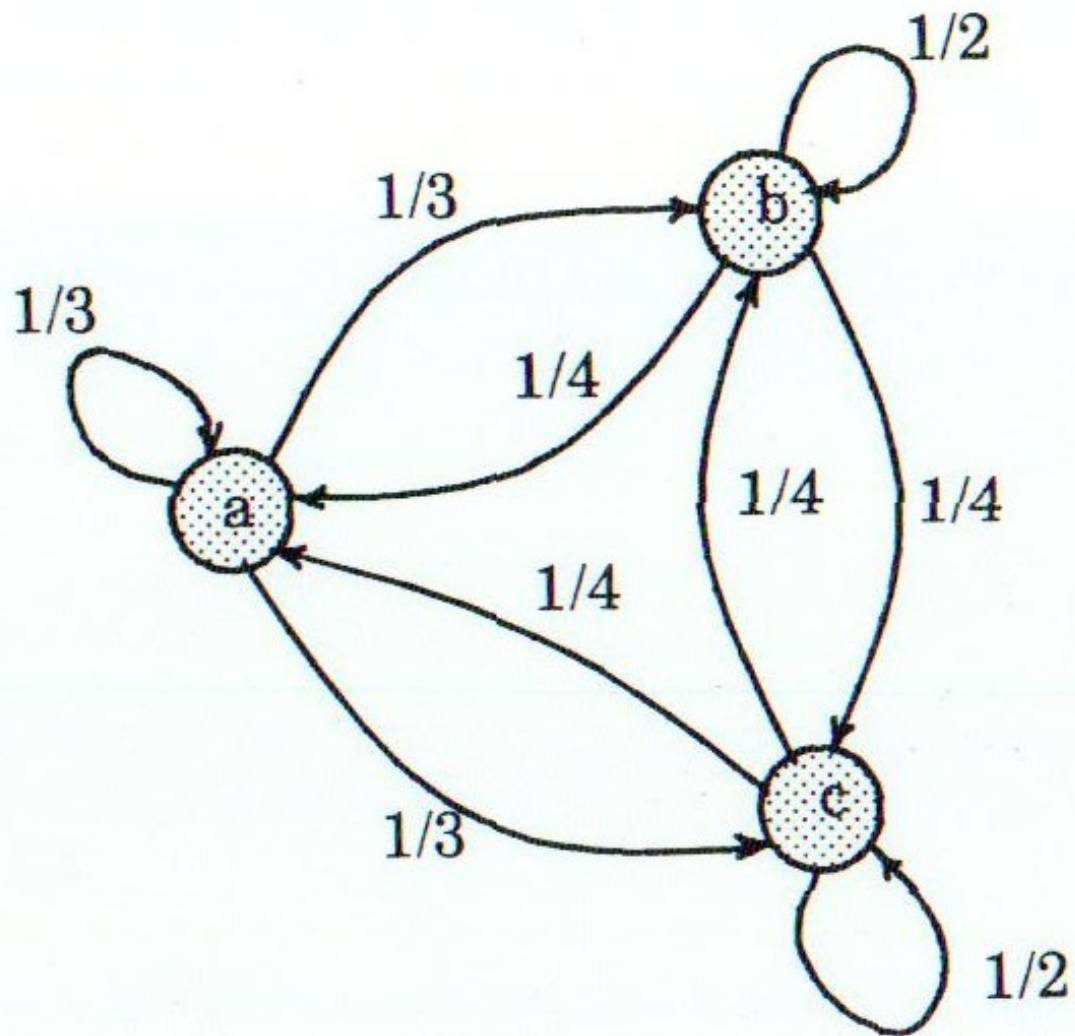


Figure 5.1.1. An example of a Markov process.

Ergodic HMM

Baum-Welch学習

シンボルが与えられたとき，尤度を最大にするように，
パラメータを学習

(初期状態確率，状態遷移確率，シンボル出力確率)

Forwardアルゴリズム
前から入力した時の尤度

Backwardアルゴリズム
後ろから入力した時の尤度

Forward-backwardアルゴリズム

HMMの学習方法

★ Baum-Walsh



大きく似たシンボルを集合させて
一つの状態にする



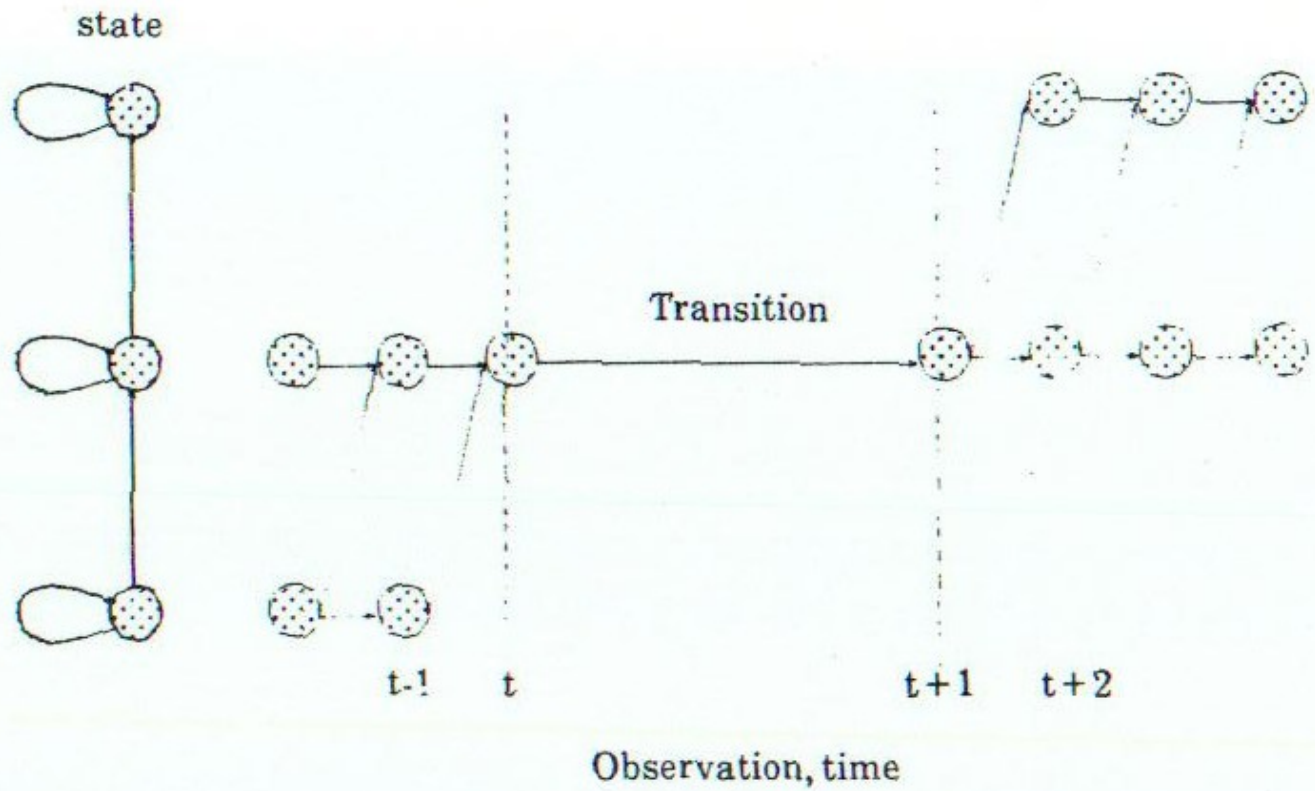


Figure 5.3.2. Illustration of computation for gamma values.

東京



大阪



京都



学習済

入力コード

A A B B C A B C

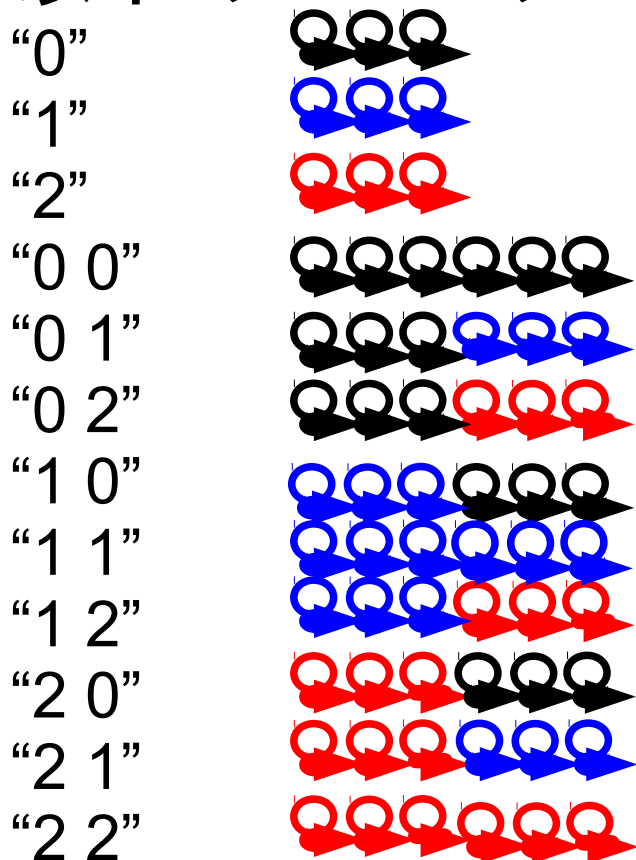
→ どのモデルが最も高い

研究のやりかた

連続単語認識

入力データ a a b b c a b c : 3単語認識

以下のHMMのモデルを作成



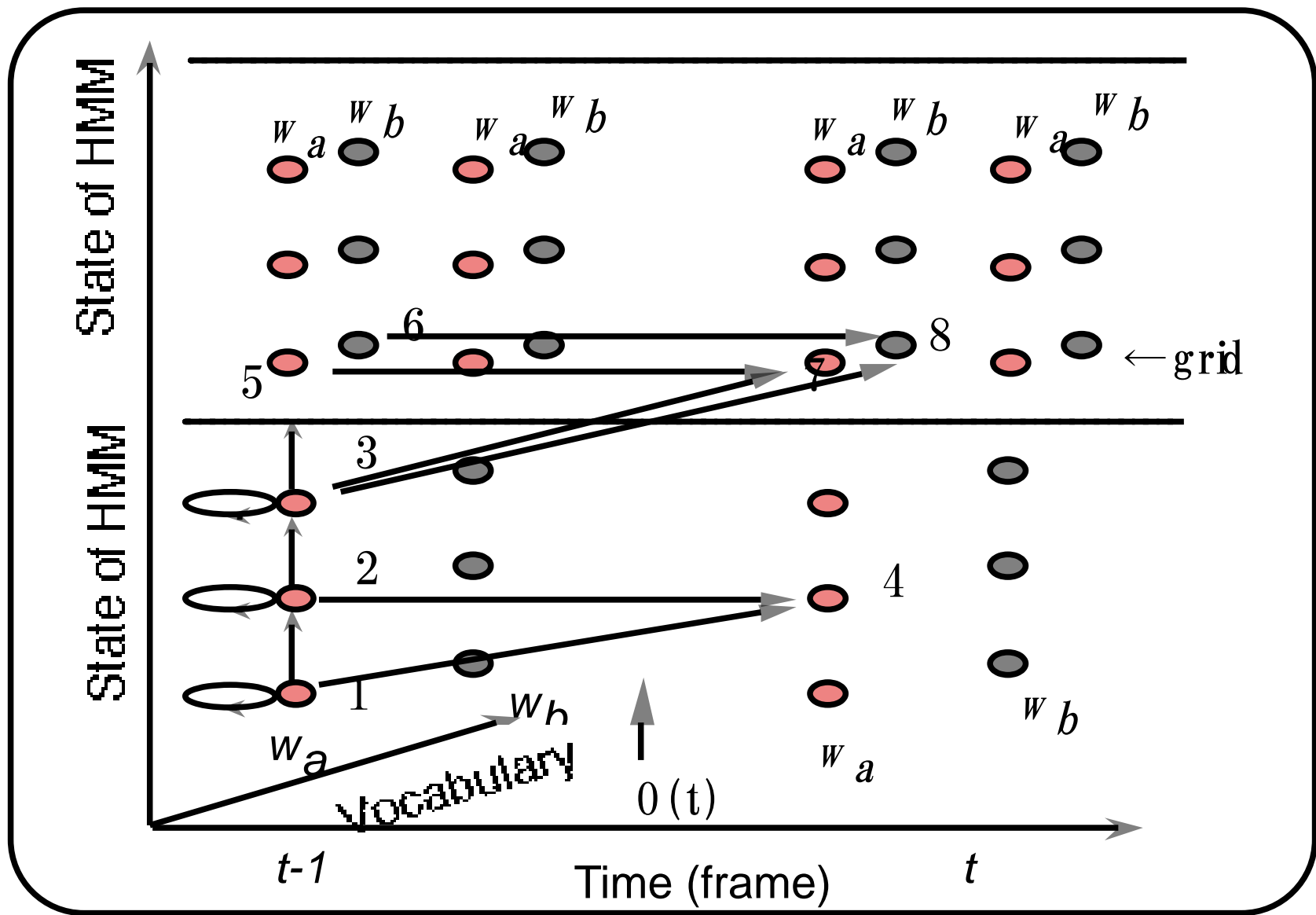
尤度のもっとも高いHMMを選択

連続音声認識のアルゴリズム

与えられたデータの尤度が最大になる
状態系列を推定

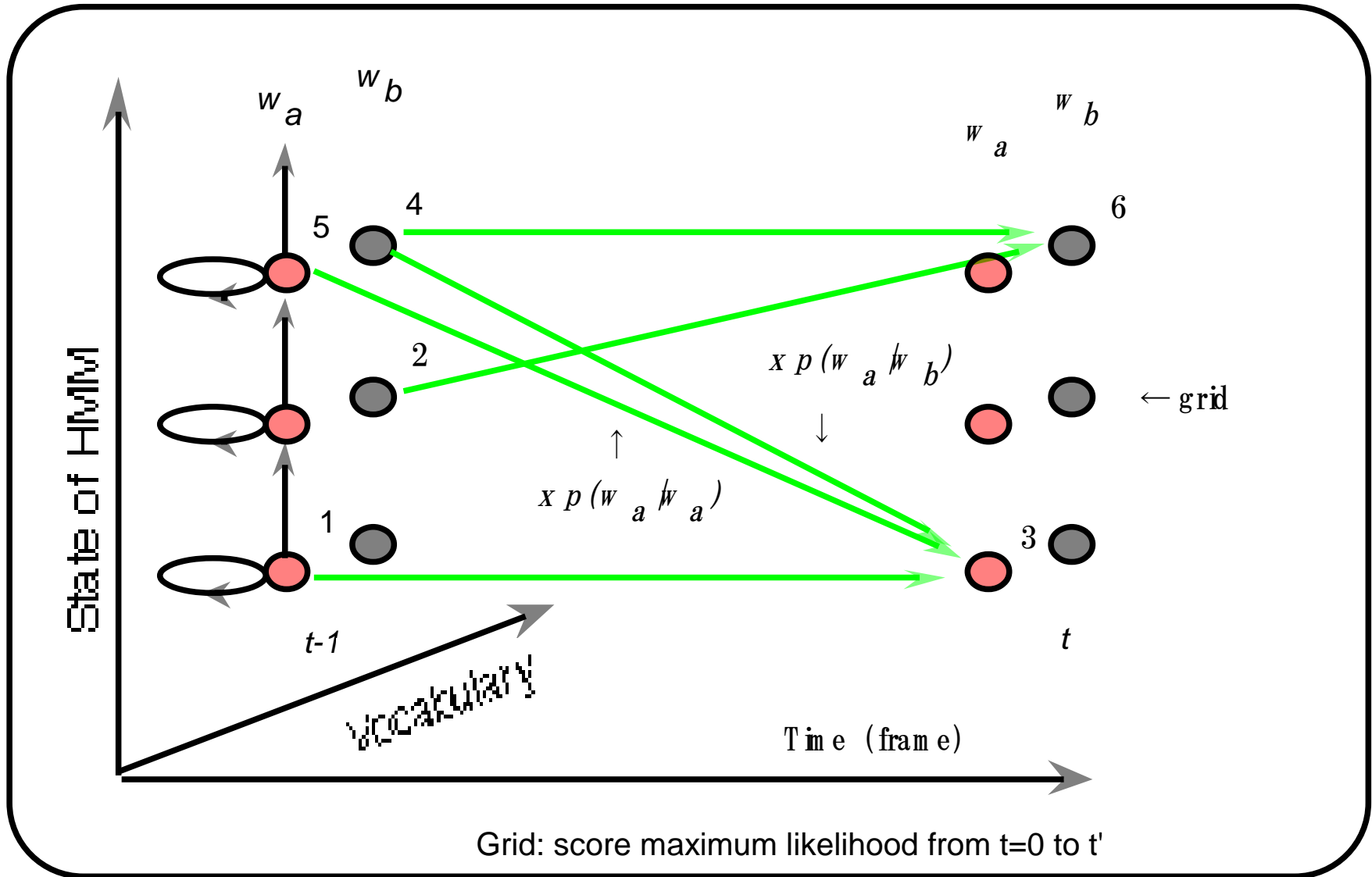
- Tree-Trellis サーチ (フルサーチ)
- Viterbiサーチ (One pass)
- (Level Building)

Tree Trellis Search

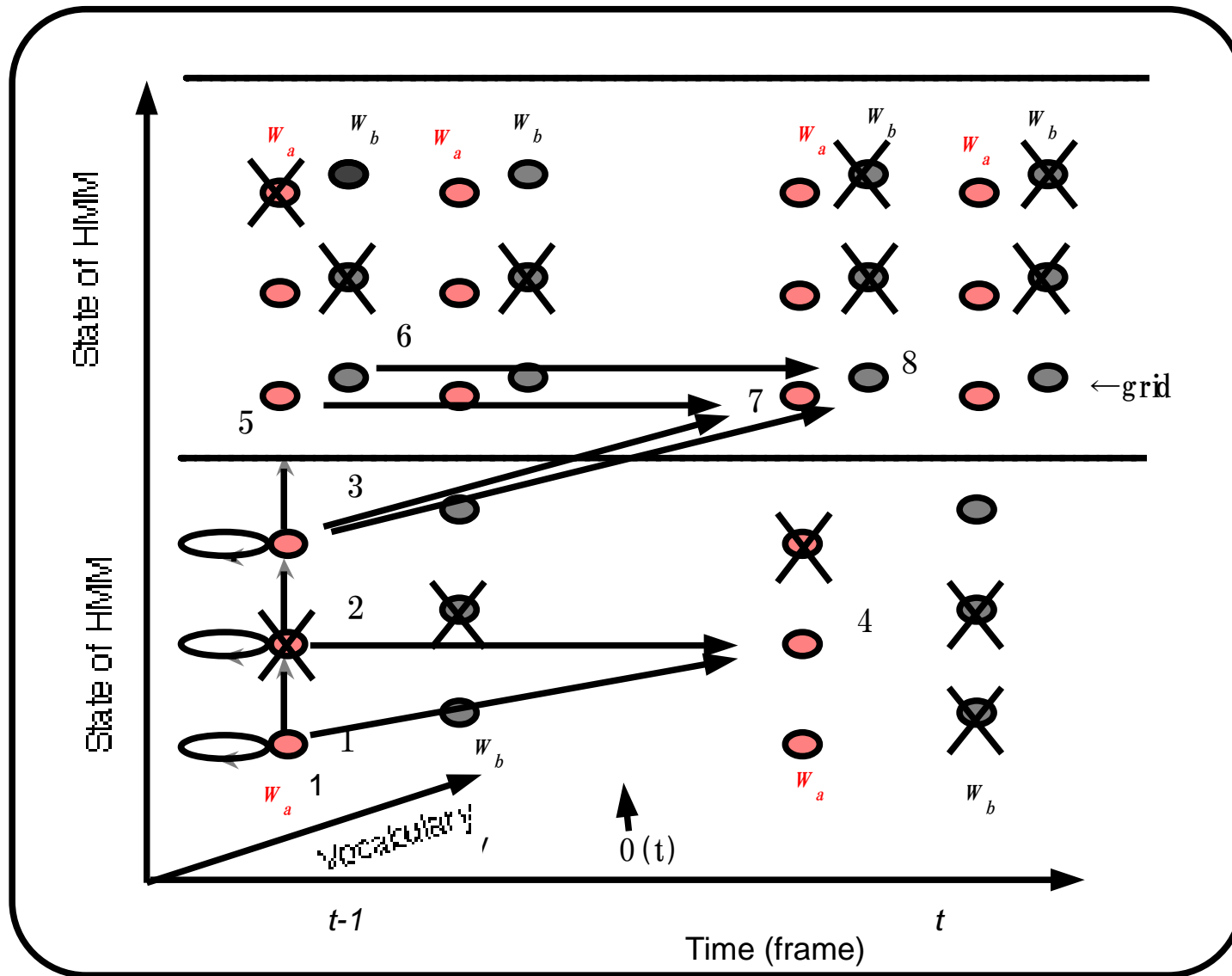


Grid: score maximum likelihood from $t=0$ to t'

One Pass DP



Beam search

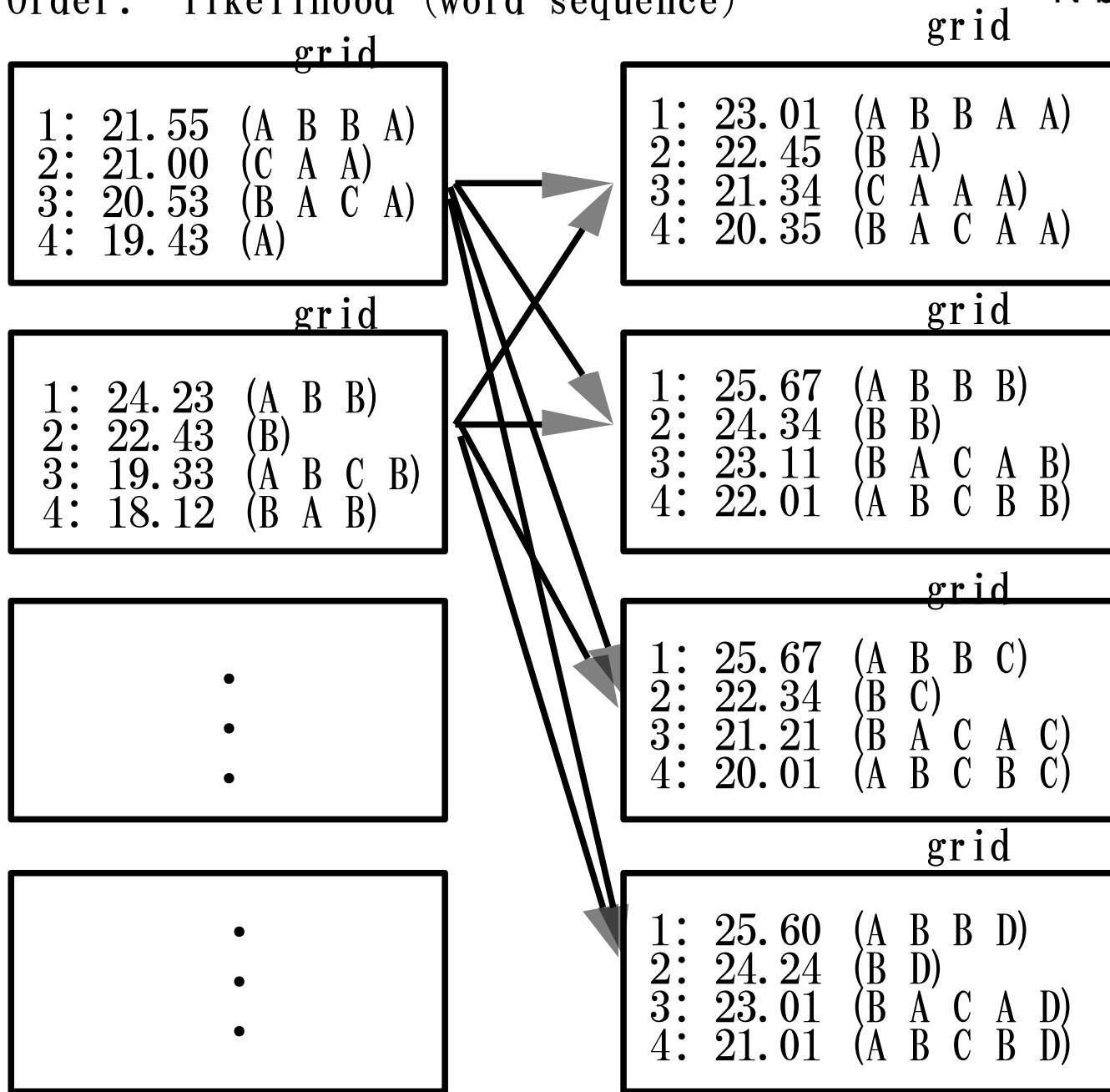


Grid: store maximum likelihood from $t=0$ to t'

X : not calculated

Order: likelihood (word sequence)

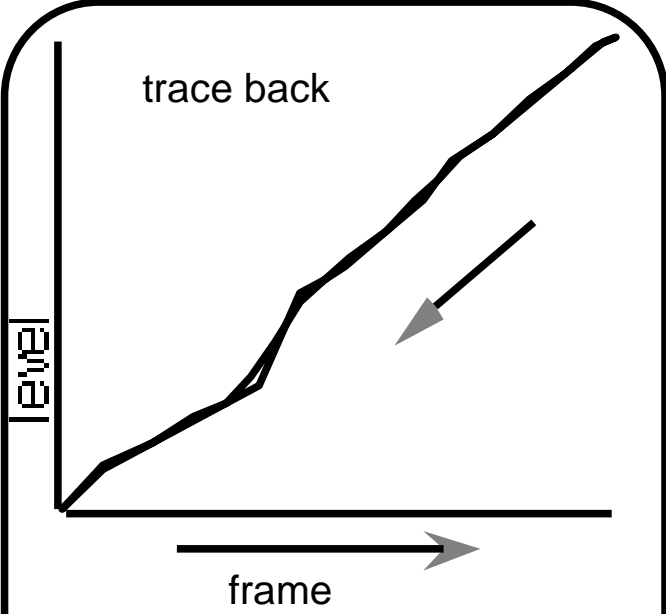
N-best



t

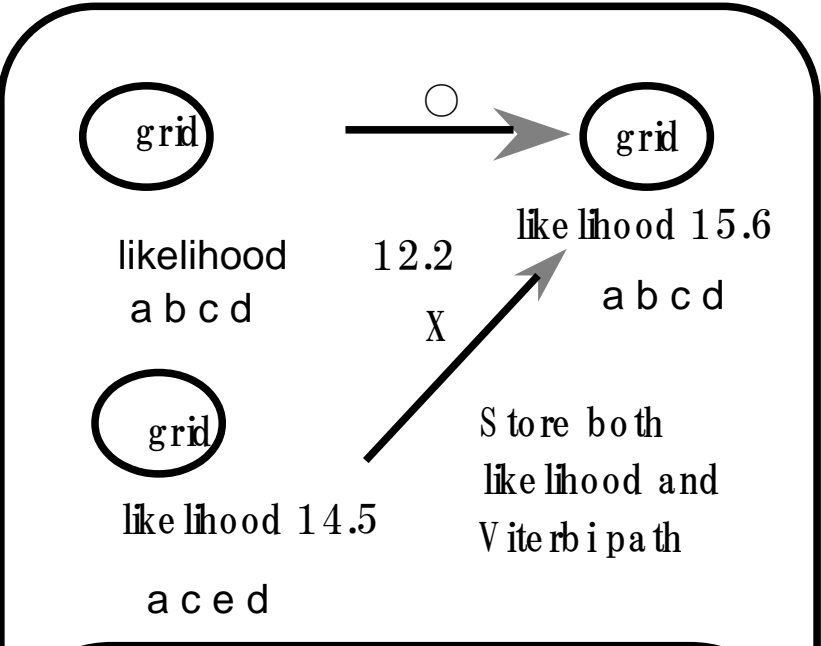
time

$t+1$



Trace back after
likelihood calculation

Normal method



Store both
likelihood and
Viterbi path

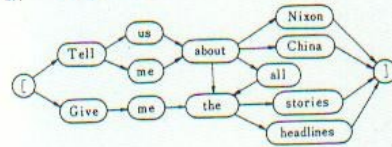
No traceback is needed

Improved method

言語処理

Harpy (arpa フォニク)

ネットワーク文法



(a) 単語を節点とするネットワーク

CFG

規則: 1. S → NP VP	I 4 : [J → の . 3]
2. NP → N J NP	[NP → NP N J . 0]
3. NP → NP N J	NP (2, 2)
4. NP → N J	[S → NP · VP . 0]
5. VP → V	[NP → NP · N J . 0]
6. N → 山	[NP → N J NP . 0]
7. N → 上	[NP → NP · N J . 2]
8. N → 木	NP → (0, 4)
9. J → の	VP (4)
a. J → を	N (4)
b. V → 切る	I 5 : [N → 木 . 4]
I 0 : [S → · NP VP . 0]	[NP → NP N · J . 0]
NP (0, 0)	[NP → NP N J . 2]
N (0)	NP (1, 4)
I 1 : [N → 山 . 0]	J (5)
NP (1, 0)	I 6 : [J → を . 5]
J (1)	[NP → NP N J . 0]
I 2 : [J → の . 1]	[NP → NP N J . 2]
NP (2, 0)	NP (2, 4)
[S → NP · VP . 0]	[S → NP · VP . 0]
[NP → NP · N J . 0]	[NP → NP · N J . 0]
NP (0, 2)	[NP → N J NP . 0]
VP (2)	[NP → NP · N J . 2]
N (2)	[NP → NP · N J . 4]
I 3 : [N → 上 . 2]	NP (0, 6)
[NP → NP N · J . 0]	[VP → V . 6]
NP (1, 2)	[V → 切る . 6]
J (3)	N (6)
I 7 : [V → 切る . 6]	I 7 : [V → 切る . 6]
[VP → V . 6]	[VP → V . 6]
[S → NP VP . 0]	[S → NP VP . 0]

但し、便宜上次の記号を用いる。NP (0, k) は右辺の三つのアイテムであるなどとし、kに対応する整数を代入するものとする。

- NP (0, k) : = [NP → · N J NP, k] N (k) : = [N → · 山, k]
- [NP → · NP N J, k] [N → · 上, k]
- [NP → · N J, k] [N → · 木, k]
- NP (1, k) : = [NP → N · J NP, k] J (k) : = [J → · の, k]
- [NP → N · J, k] [J → · を, k]
- NP (2, k) : = [NP → N J · NP, k]
- [NP → N J ·, k]

図 5.9 例文「山の上の木を切る。」のパースタテブル

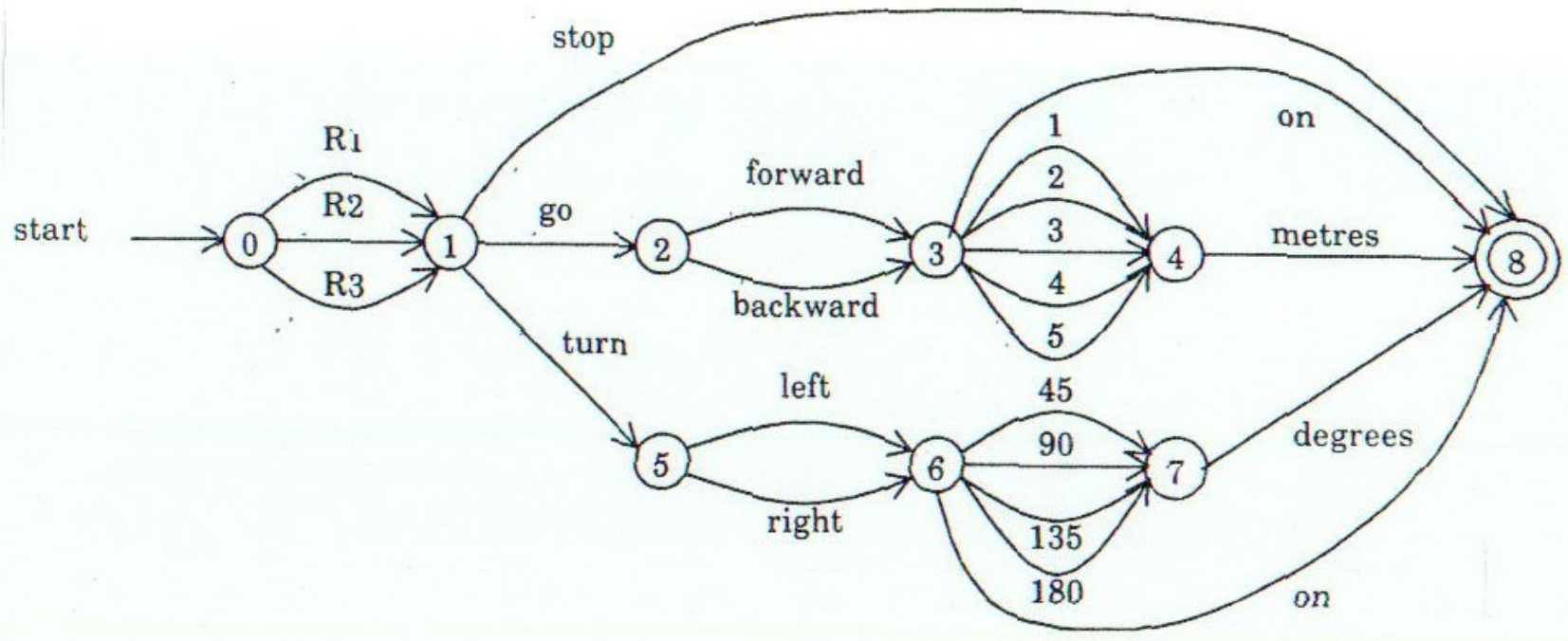


Figure 3.3.2. Example of finite state automata.

自然言語

人間と人間のインターフェース
日本語、英語

人工言語

人間とコンピュータのインターフェース
C、Fortran、Basic

自然言語処理

自然言語でコンピュータとインターフェース

例

仮名漢字変換 機械翻訳 対話処理
検索 その他



チューリングマシン

文脈依存文法
(Context Sensitive Grammar)

文脈自由文法
(Context Free Grammar)

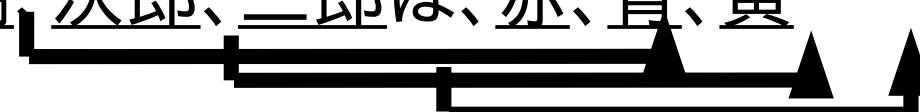
正規文法
(Network grammar)

チョムスキーの言語階層

チューリングマシン

文脈依存文法

一郎、次郎、三郎は、赤、青、黄
のテープを取った。

A diagram of a Turing machine tape. The tape contains the text "一郎、次郎、三郎は、赤、青、黄" followed by a period. Below the text, a horizontal line represents the tape. Three upward-pointing arrows are positioned above the line, indicating the head's position at different points: one under "三郎は", one under "青", and one under "黄".

(Context Sensitive Grammar)

文脈自由文法

東京にいる人に会いにいった。

A diagram of a Turing machine tape. The tape contains the text "東京にいる人に会いにいった。". Below the text, a horizontal line represents the tape. One upward-pointing arrow is positioned above the line, indicating the head's position under "に".

(Context Free Grammar) 複文

正規文法

(Network grammar)

私は 山に いった。
単文

自然言語処理の基本

自然言語処理パーザー

プログラム + 文法規則 + 単語辞書

受理

正規文法

有限状態オートマトン

非終端記号 A, B, C, \dots (品詞)

終端記号 a, b, c, \dots (単語)

$A \rightarrow B$

$B \rightarrow b$

自然言語処理の基本

形態素解析、構文解析

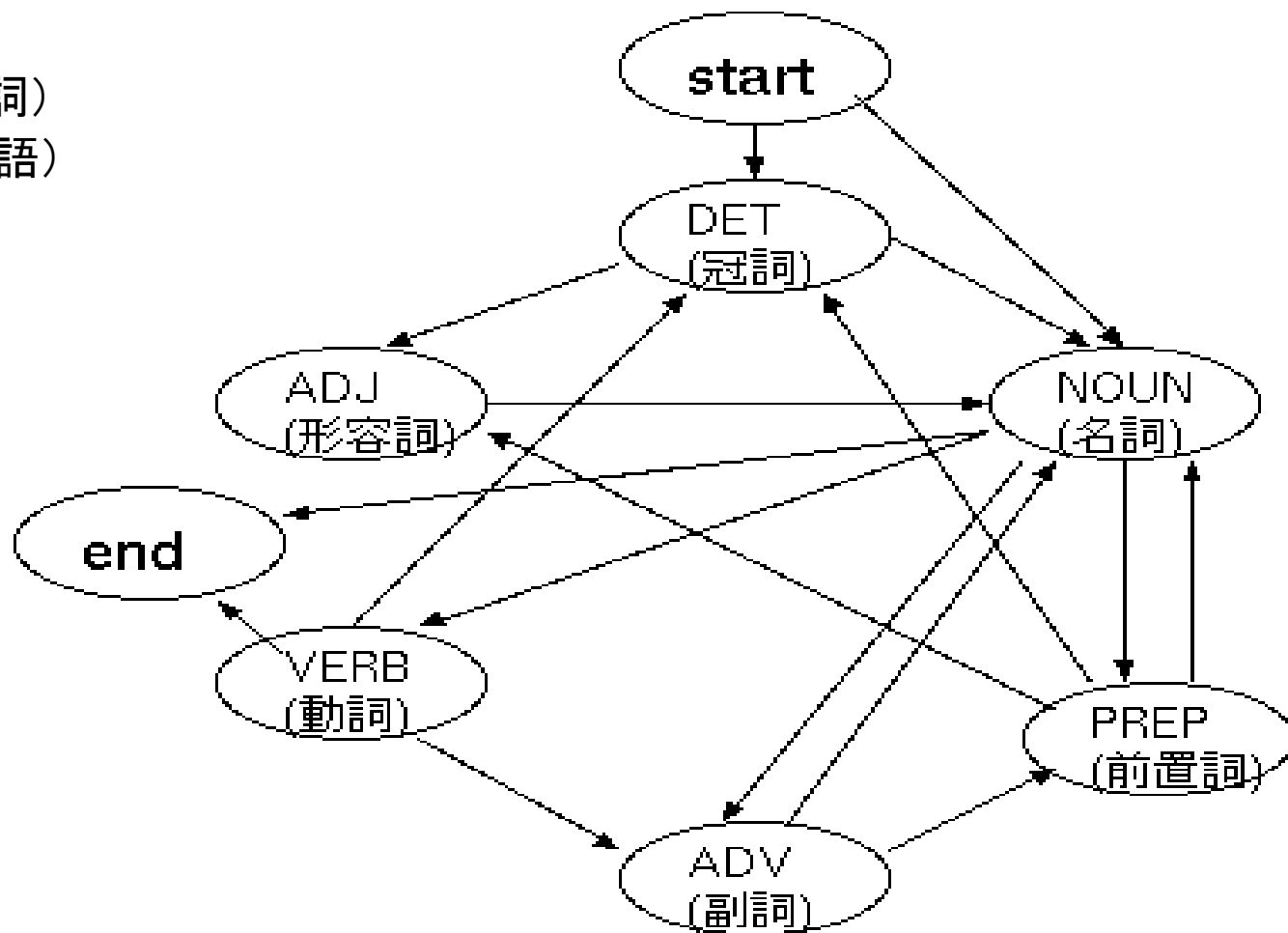
形態素解析

	the	child	runs	quickly	to	the	large	house	
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ	NOUN	end
初期状態	冠詞	名詞	動詞	副詞	前置詞	冠詞	形容詞	名詞	最終状態

A→B
B→a

A,B,C 非終端記号 (品詞)
a 終端記号 (単語)

文法規則



有限状態オートマトン

ネットワーク文法

文法規則		単語辞書	
start	→ DET (冠詞)		
start	→ NOUN (名詞)		
DET(冠詞)	→ ADJ(形容詞)		
DET(冠詞)	→ NOUN(名詞)		
ADJ(形容詞)	→ NOUN(名詞)		
NOUN(名詞)	→ PREP(前置詞)	the	→ DET(冠詞)
PREP(前置詞)	→ NOUN (名詞)	child	→ NOUN(名詞)
PREP(前置詞)	→ ADJ(形容詞)	runs	→ VERB(動詞)
PREP(前置詞)	→ DET(冠詞)	quickly	→ ADV(副詞)
NOUN(名詞)	→ VERB(動詞)	to	→ PREP(前置詞)
VERB(動詞)	→ ADV(副詞)	the	→ DET(冠詞)
VERB(動詞)	→ DET(冠詞)	large	→ ADJ(形容詞)
ADV(副詞)	→ PREP(前置詞)	house	→ NOUN(名詞)
NOUN(名詞)	→ ADV(副詞)		
ADV(副詞)	→ NOUN(名詞)		
VERB(動詞)	→ end		
NOUN(名詞)	→ end		

表 1: NetWork 文法の動作

start											
		the									
start	DET										
		the	child								
start	DET	NOUN									
		the	child	runs							
start	DET	NOUN	VERB								
		the	child	runs	quickly						
start	DET	NOUN	VERB	ADV							
		the	child	runs	quickly	to					
start	DET	NOUN	VERB	ADV	PREP						
		the	child	runs	quickly	to	the				
start	DET	NOUN	VERB	ADV	PREP	DET					
		the	child	runs	quickly	to	the	large			
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ				
		the	child	runs	quickly	to	the	large	house		
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ	NOUN			
		the	child	runs	quickly	to	the	large	house	.	
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ	NOUN	end		

network文法の解析

文脈自由文法 (CFG)

PushDownオートマトン

非終端記号 A,B,C... (品詞)

終端記号 a,b,c.. (単語)

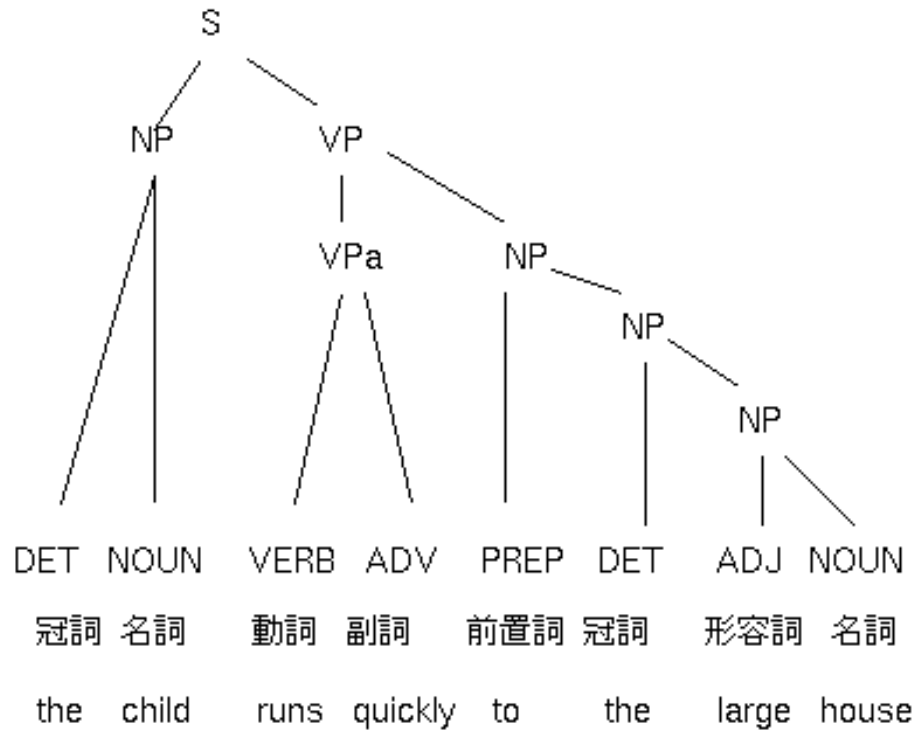
$A \rightarrow B + C$

$B \rightarrow b$

$C \rightarrow c$

構文解析

文脈自由文法



$A \rightarrow B + C$

$B \rightarrow a$

A, B, C 非終端記号
(品詞)

a 終端記号
(単語)

文脈自由文法

文法規則	単語辞書
SENTENCE(文) → NP(名詞句) + VP(動詞句)	
NP(名詞句) → DET(冠詞) + NOUN(名詞句)	the → DET(冠詞)
NP(名詞句) → DET(冠詞) + NP(名詞句)	child → NOUN(名詞)
NP(名詞句) → PREP(前置詞) + NP(名詞句)	runs → VERB(動詞)
NP(名詞句) → ADJ(形容詞) + NOUN(名詞)	quickly → ADV(副詞)
VP(動詞句) → VERB(動詞) + NP(名詞句)	to → PREP(前置詞)
VP(動詞句) → VP1(動詞句) + NP(名詞句)	the → DET(冠詞)
VP(動詞句) → VERB(動詞) + ADV(副詞)	large → ADJ(形容詞)
VP(動詞句) → VERB(動詞)	house → NOUN(名詞)
VP _a (動詞句) → VERB(動詞) + ADV(副詞)	
VP _a (動詞句) → VERB(動詞) + NP(名詞句)	

構文解析方法

Top Down 法

CYK 法

Early 法

LR 法

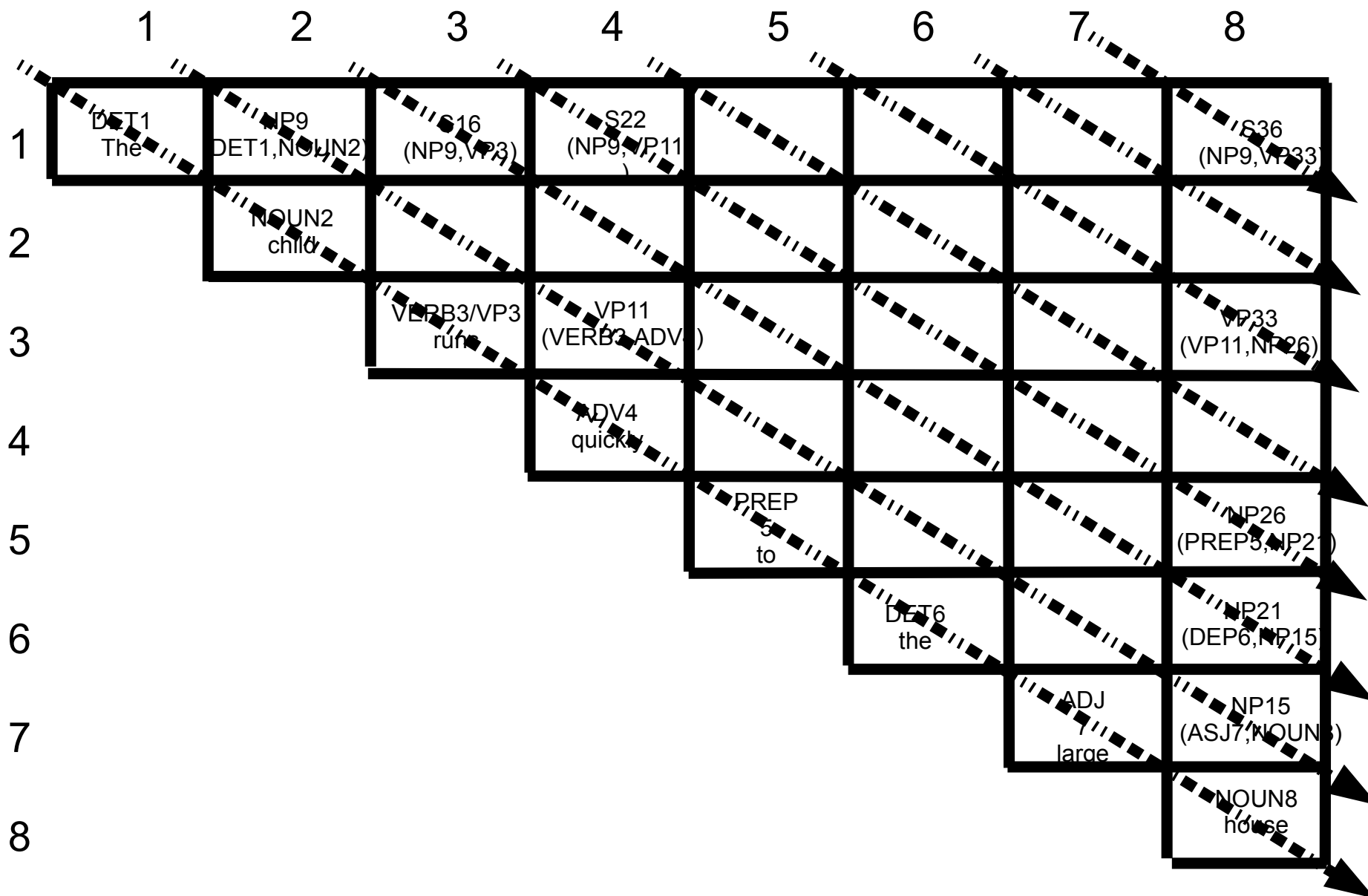
Top Down 法のフロー

1	S_1	→	$NP_1 + VP_1$	
2	NP_1	→	$DET_2 + NOUN_2$	1のNPを処理する.
3	DET_2	→	the	
4	$NOUN_2$	→	child	1のNPが受理されたのでVPを処理する.
5	VP_1	→	$VERB_5 + NP_5$	
6	$VERB_5$	→	runs	
7	NP_5	→	$DET_7 + NOUN_7$	
8	DET_7	→	quickly	
9	NP_5	→	$DET_7 + NP_7$	
10	DET_7	→	quickly	
11	NP_5	→	$PREP_{11} + NP_{11}$	
12	$PREP_{11}$	→	quickly	
13	NP_5	→	$ADJ_{13} + NOUN_{13}$	
14	ADJ_{13}	→	quickly	
15	VP_1	→	$VP_{15} + NP_{15}$	
16	VP_{15}	→	$VERB_{16} + ADV_{16}$	
17	$VERB_{16}$	→	runs	
18	ADV_{16}	→	quickly	
19	NP_{15}	→	$DET_{19} + NOUN_{19}$	
20	DET_{19}	→	to	
21	NP_{15}	→	$DET_{21} + NP_{21}$	
22	DET_{19}	→	to	
23	NP_{15}	→	$PREP_{23} + NP_{23}$	
24	$PREP_{23}$	→	to	24が受理
25	NP_{23}	→	$DET_{25} + NOUN_{25}$	
26	DET_{25}	→	the	
27	$NOUN_{25}$	→	large	26が受理されず.
28	NP_{23}	→	$DET_{28} + NP_{28}$	
29	DET_{28}	→	the	25が受理
30	NP_{28}	→	$DET_{30} + NOUN_{30}$	
31	DET_{30}	→	large	
32	NP_{28}	→	$DET_{32} + NP_{32}$	
33	DET_{32}	→	large	
34	NP_{32}	→	$PREP_{34} + NP_{34}$	
35	$PREP_{34}$	→	large	
36	NP_{32}	→	$ADJ_{36} + NOUN_{36}$	
37	ADJ_{36}	→	large	
38	$NOUN_{36}$	→	house	

Top Down 法 (ひたすらルールを適応)

Top-Downパーサでは記述してはいけない文法例
(左再帰規則)

VP(動詞句) → VP(動詞句) + NP(名詞句)



CYK法

CYKの文法規則

$A_{i,j}$ には $A_{i,k} A_{k+1,j}$ の書き換え規則

例

$$\begin{aligned} A_{3,7} &= A_{3,3} A_{4,7} \\ &= A_{3,4} A_{5,7} \\ &= A_{3,5} A_{6,7} \\ &= A_{3,6} A_{7,7} \end{aligned}$$

考察の内容

- 2) I saw a girl with a telescope には大きくわけて2つの意味(解釈)がある. このときの構文解析結果と, 日本語の訳
- 4) Times flies like an arrow をコンピュータで解釈した場合の多義数

N-gram model

Unigram

$$p(w_i)$$

モデルLにおける単語 w_i の出現頻度

Bigram

$$p(w_j | w_i)$$

w_i 頻が出現したときに
単語 w_j に遷移する確率

Trigram

$$p(w_k | w_i w_j)$$

単語 $w_i w_j$ が出現したとき
単語 w_k に遷移する確率

Trigramを越える言語モデルの研究

1. trigramモデルの長点

簡単
学習が容易
出力がreasonable

2. trigramモデルの短所

学習データが少ないときの連鎖確率の信頼性
重文, 複文などの依存構造を持たない

3. 文法を持たない.

4. 意味的におかしい文が出力される.