14. 連続音声認識に挑戦しよう

- 音声認識の原理

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{x}) = \arg\max_{\boldsymbol{w}} P(\boldsymbol{x}|\boldsymbol{w})P(\boldsymbol{w})$$

  - 入力 $\boldsymbol{x}$ のもとで事後確率 $P(\boldsymbol{w}\,|\,\boldsymbol{x})$ を最大にする単語列 $\hat{\boldsymbol{w}}$ を認識結果とする
  - $P(\boldsymbol{x}|\boldsymbol{w})$: 音響モデル ...HMM を用いて計算
  - $P(\boldsymbol{w})$: 言語モデル ...N-gram を用いて計算
- 問題点
  - 大語彙（数千語以上）の場合、全ての可能な $\boldsymbol{w}$ をリストアップすることは不可能

14.1 基本的な探索手法

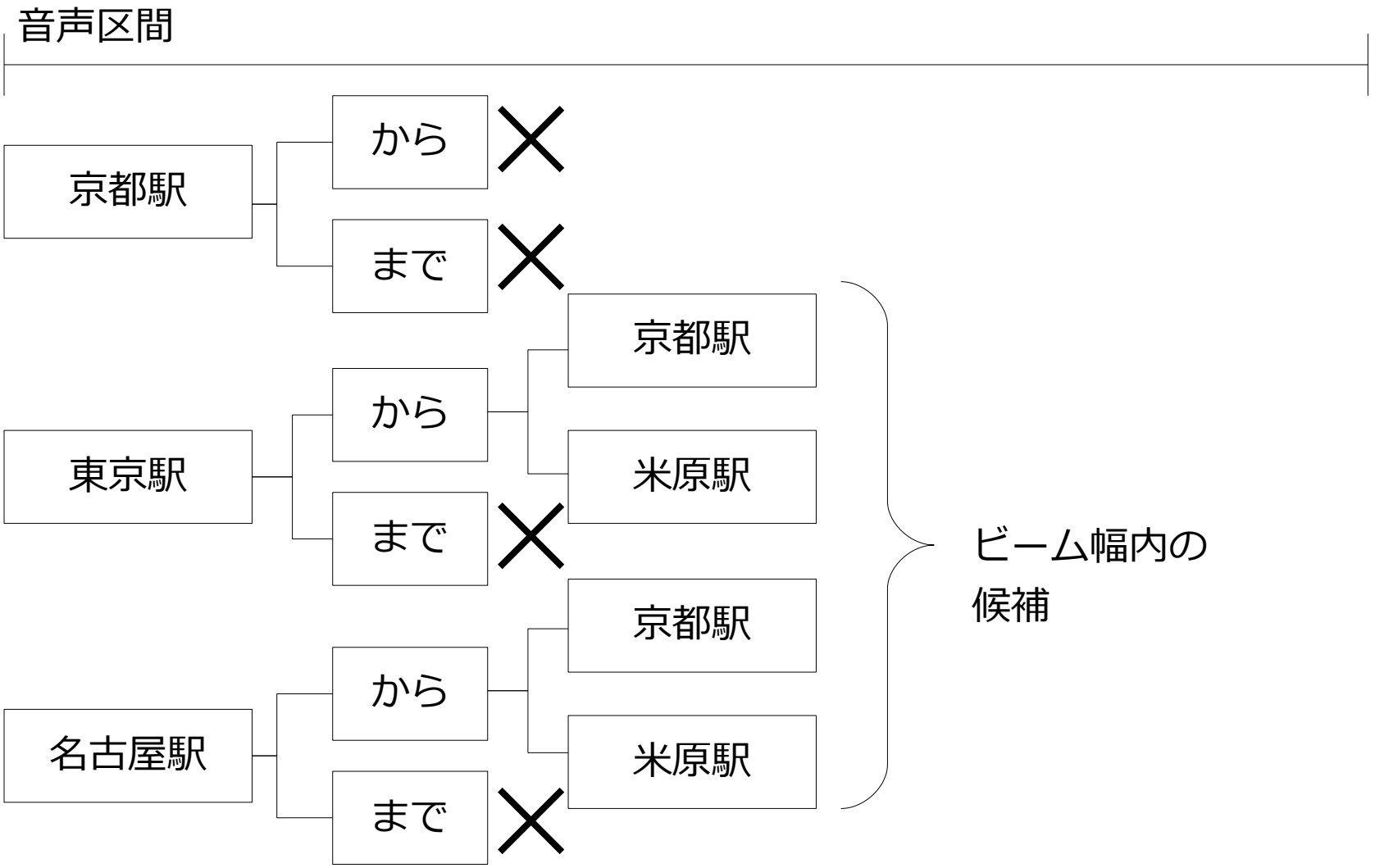
- 探索の導入
  - 解候補となる仮説を動的に展開することで、効率よく解（事後確率最大の単語列）を見つける方法
  - 単純な探索の例：縦型探索

14.1 基本的な探索手法

- 最適解を求めて：解の絞り方
  - 探索幅を制限する ... ビームサーチ
  - 評価値の高い候補を優先する
    ... ヒューリスティックサーチ
  - 探索空間を静的に展開し、最適化 ...WFST

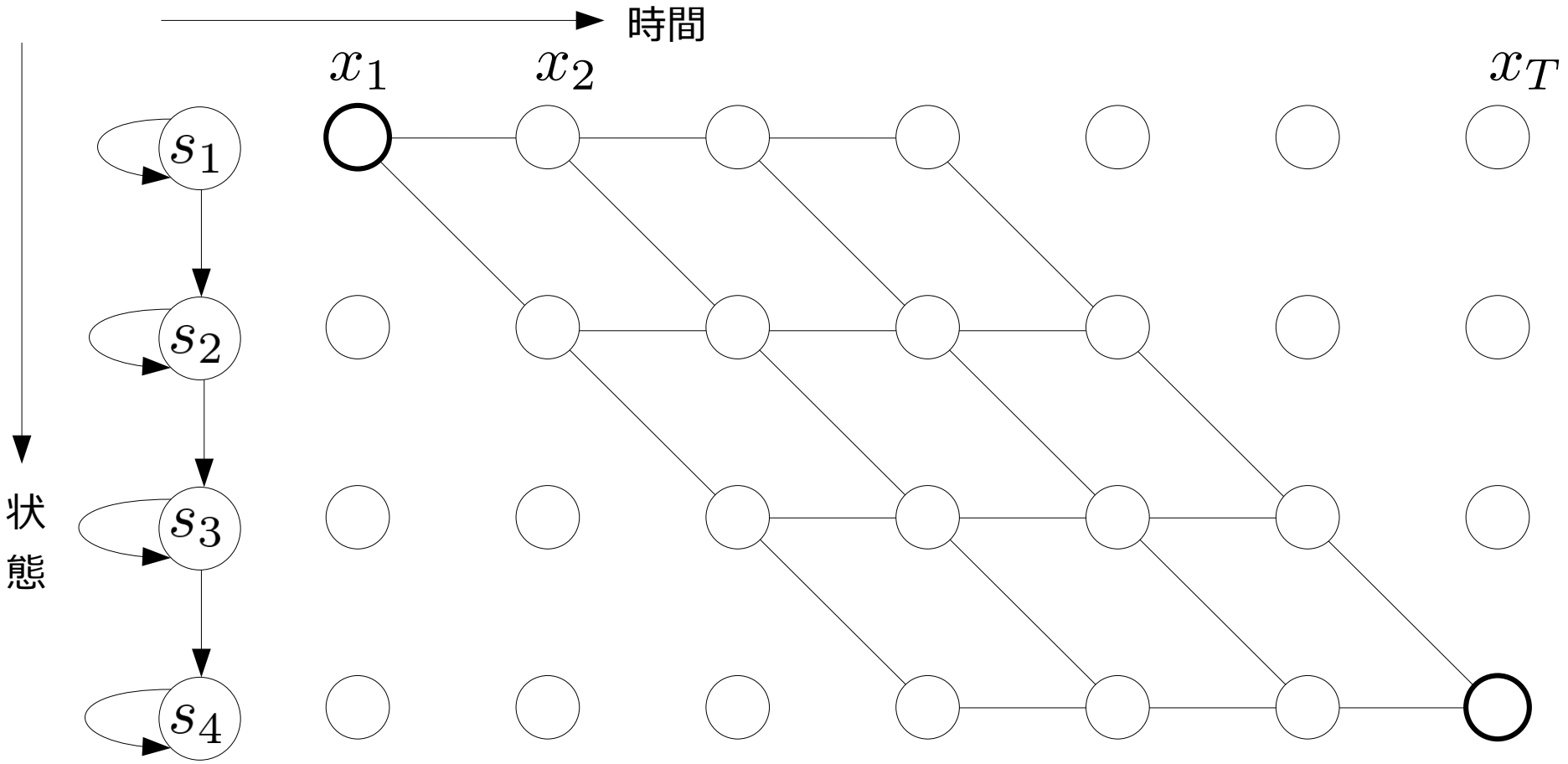
14.1 基本的な探索手法

- ビームサーチとは
  - 探索の幅を一定数のスコアの高い候補に限定



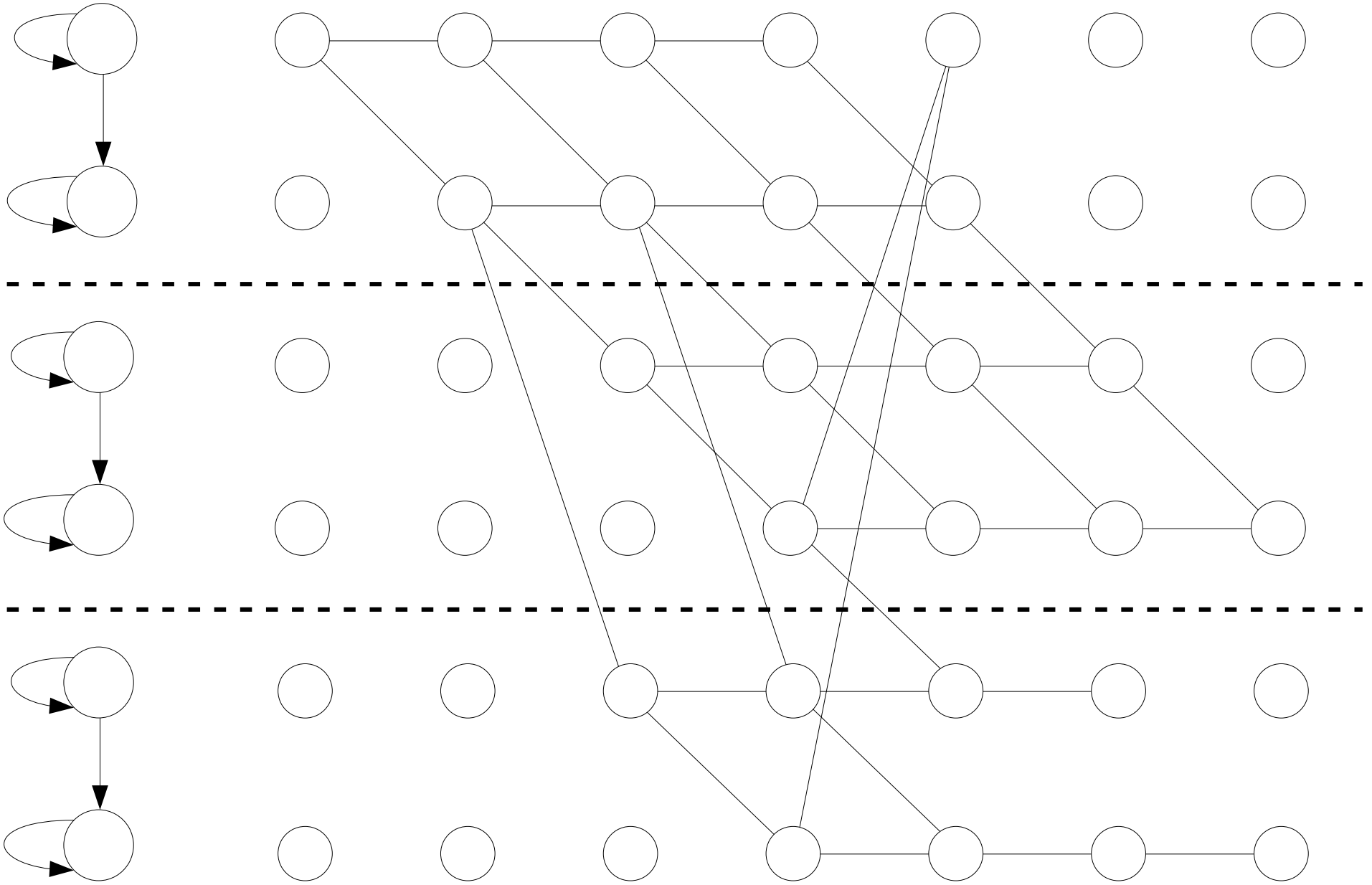
14.1 基本的な探索手法

- 音声認識ではトレリス空間上で実現



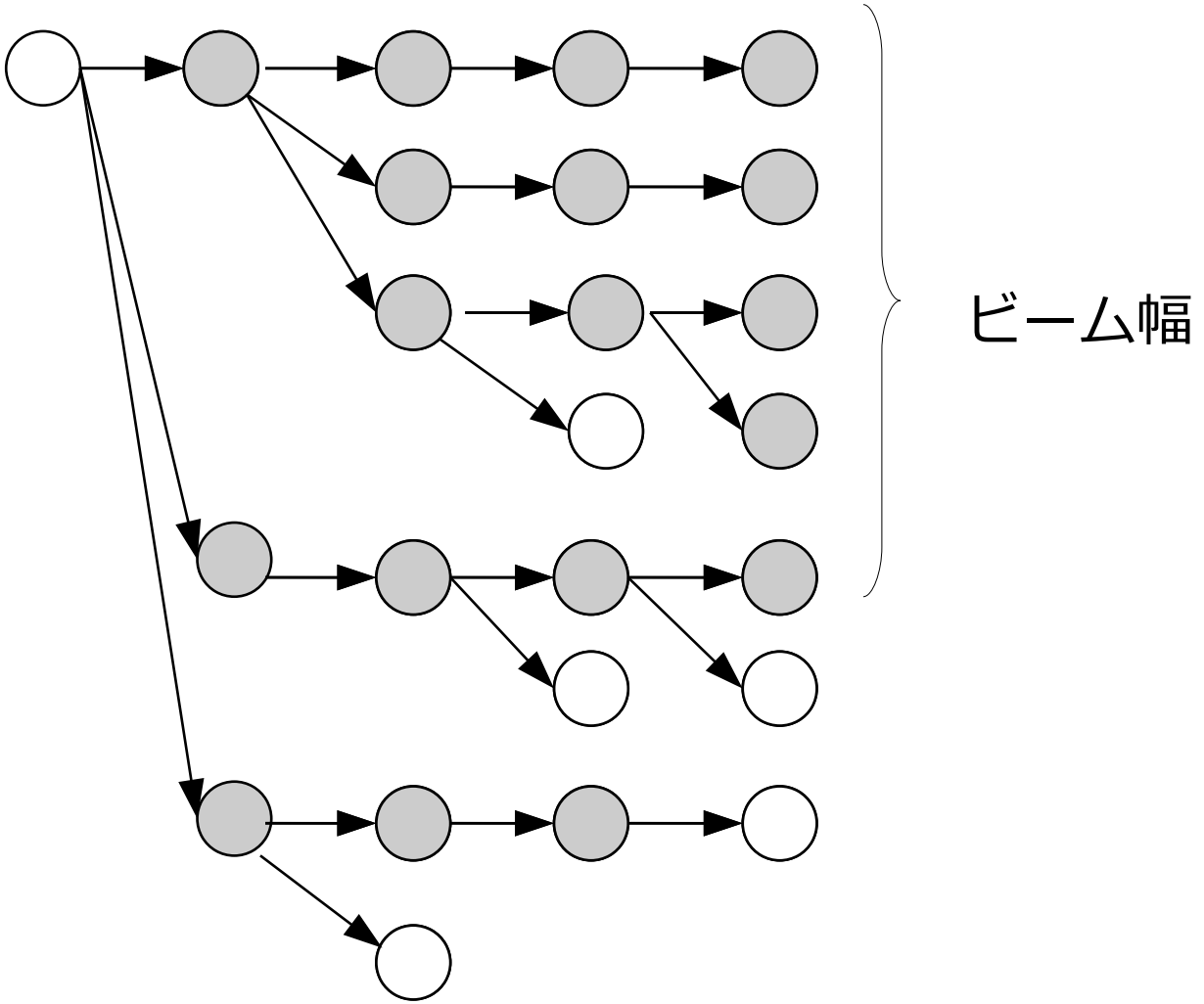
14.1 基本的な探索手法

- 音素 HMM を結合したトレリス空間の例



14.1 基本的な探索手法

- トレリスに対する探索幅（ビーム）の導入

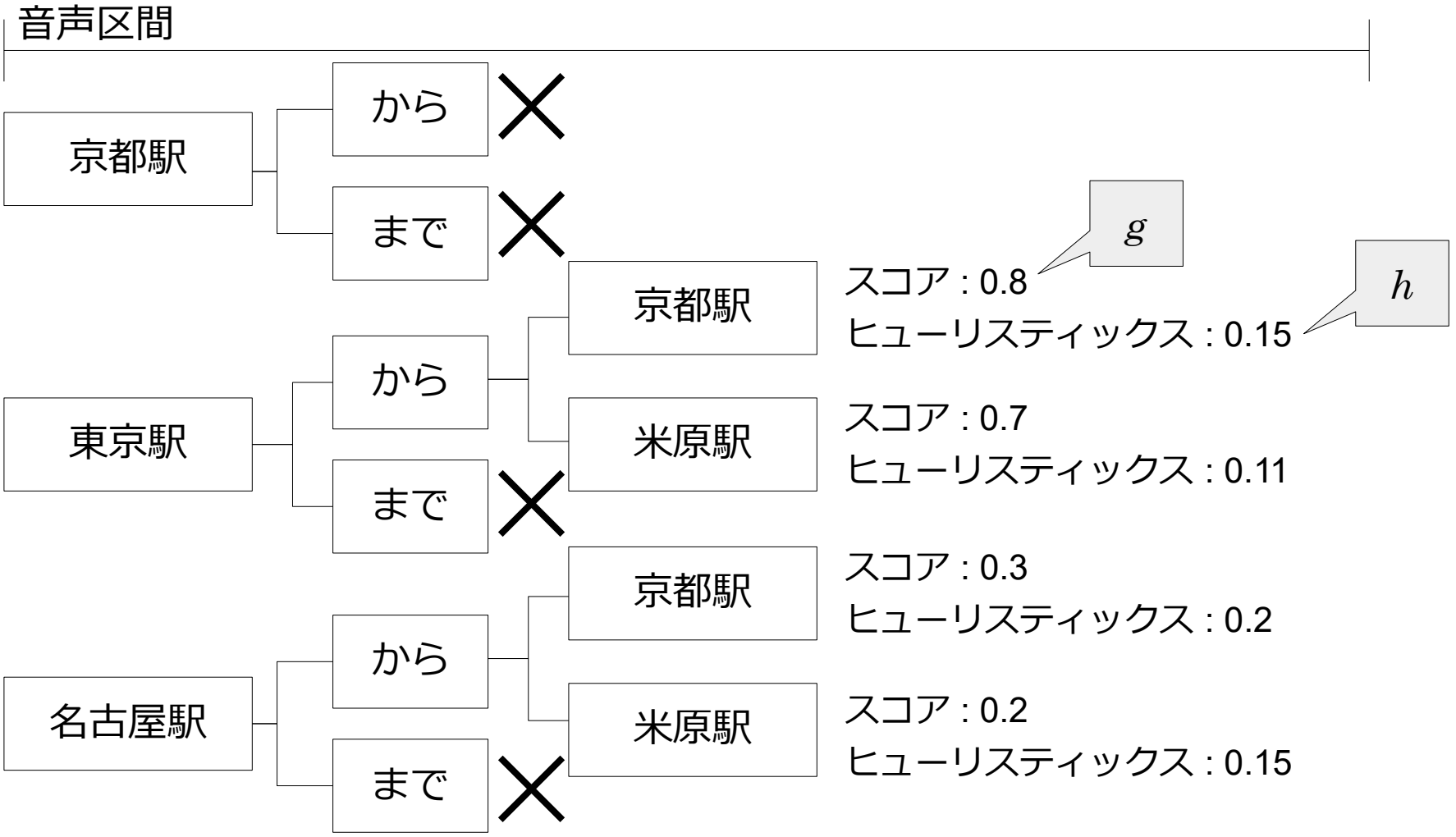


14.2 ヒューリスティック探索

- ビームサーチの問題点
  - 適切なビーム幅は？
    - 広すぎると、計算量が増大
    - 狭すぎると、正解を落としてしまう確率が高まる
  - 言語モデルスコアの探索への組み込み
    - すべての音響スコアの計算が終わってからでは非効率
    - 単語境界でスコアを組み込むと、そのフレームで単語境界となっていない候補との間で不公平な評価になる

14.2 ヒューリスティック探索

- ヒューリスティックサーチとは
  - 各候補の**今後の**スコア（ヒューリスティックス）を予測
  - 最良優先探索： $g + h$ の高い順に探索

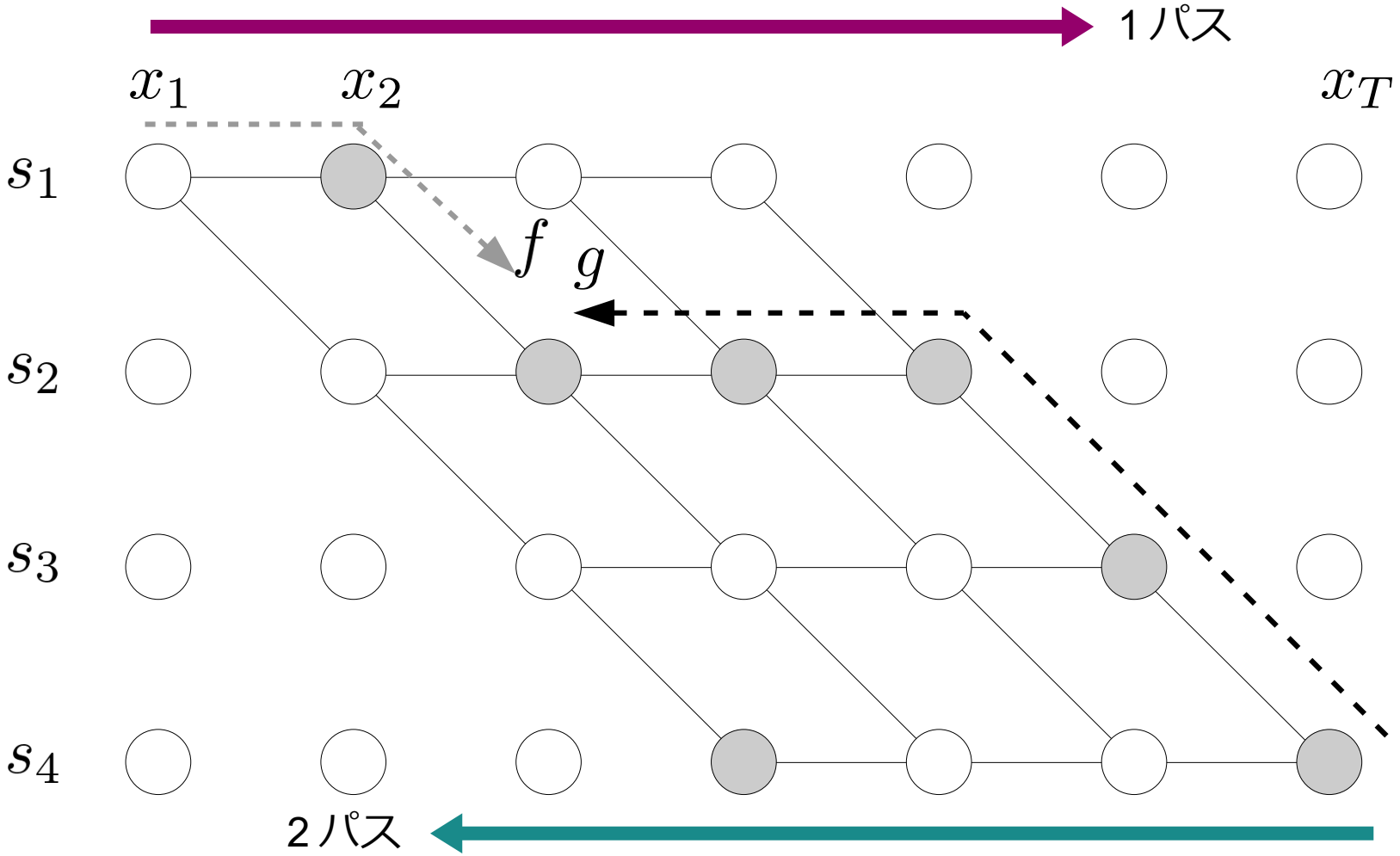


14.2 ヒューリスティック探索

- 今後のスコアの求め方
  - 2 回探索を行う
    - まず高速に動作する粗い計算でおおよその確率を求める
    - 次に精密な計算でヒューリスティックサーチを行う

14.2 ヒューリスティック探索

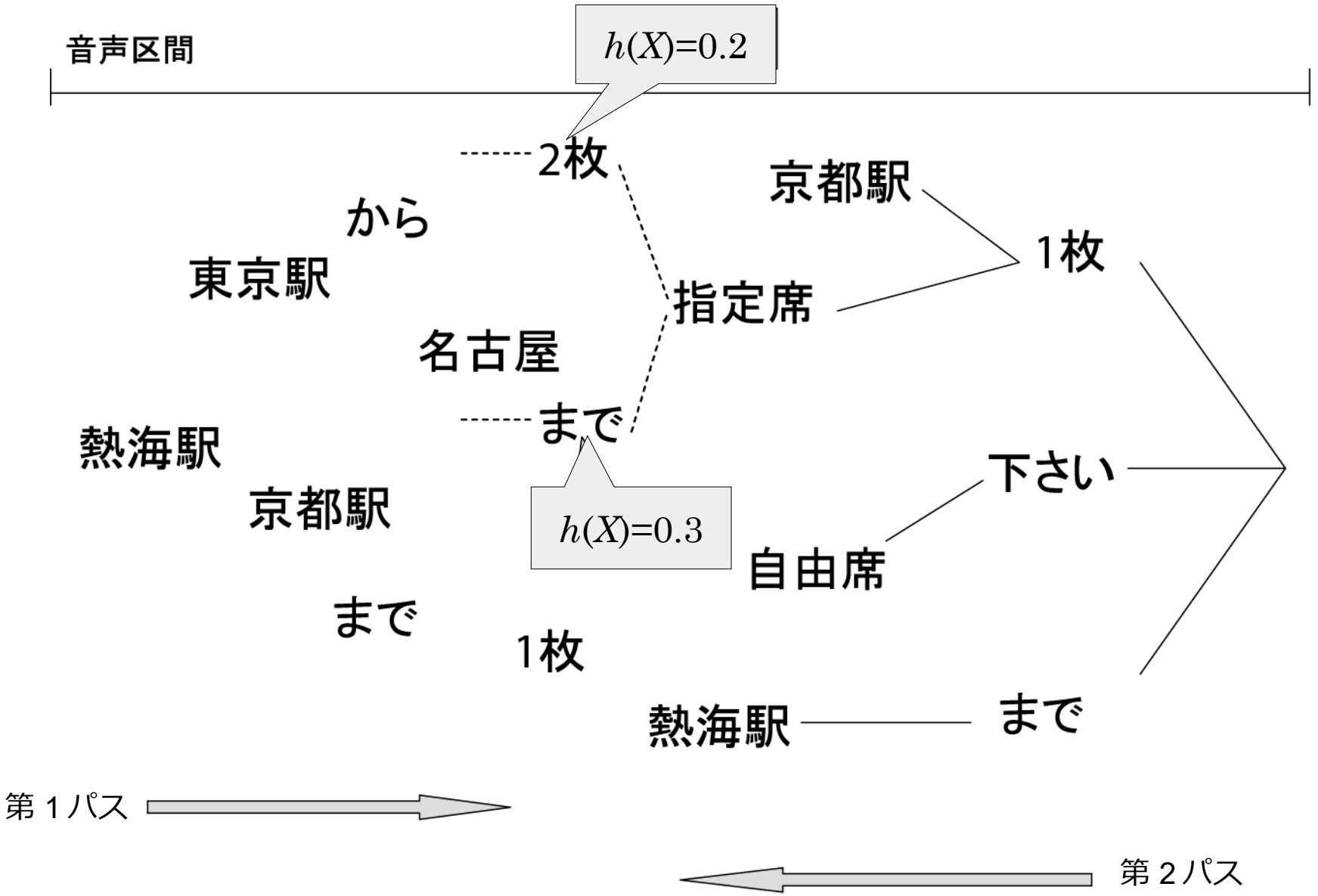
- ヒューリスティックスを用いた 2 パス探索



$$\hat{h} = f + g$$

14.2 ヒューリスティック探索

- Julius における 2 パスサーチ

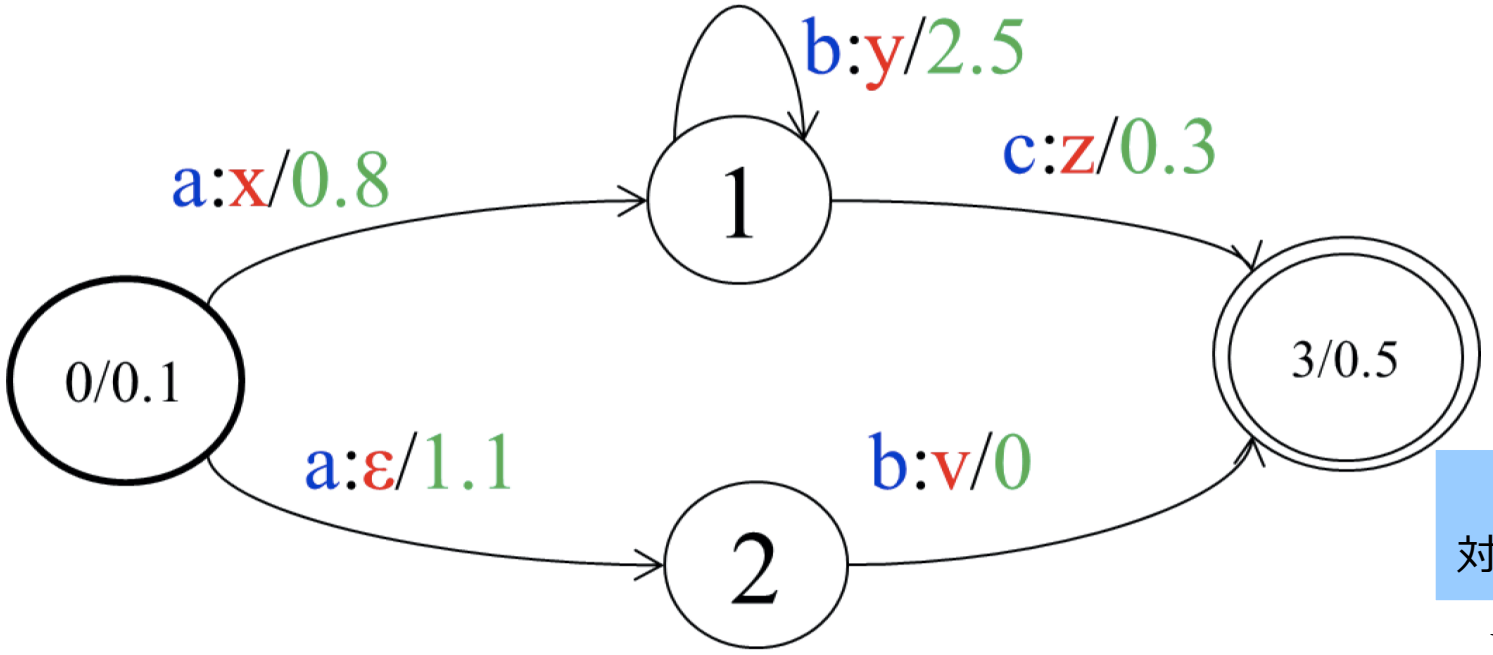


14.2 ヒューリスティック探索

- Julius における 2 パスサーチ
  - 第 1 パス（フレーム同期ビーム探索）
    - 言語モデルは 2 グラム
    - 単語間の音素変形は考慮しない
    - 出力は単語トレリス（言語モデルスコアの公平な利用）
  - 第 2 パス（スタックデコーディング）
    - 第 1 パスの結果を逆方向に見てヒューリスティックスとする
    - 言語モデルは 3 グラム
    - 単語間にも triphone を適用

14.3 WFST による探索手法

- WFST とは
  - Weighted Finite State Transducer （重み付き有限状態トランスデューサ）
  - 記号列を入力し、別の記号列と重みを出力



入力 ac に
対する出力は？

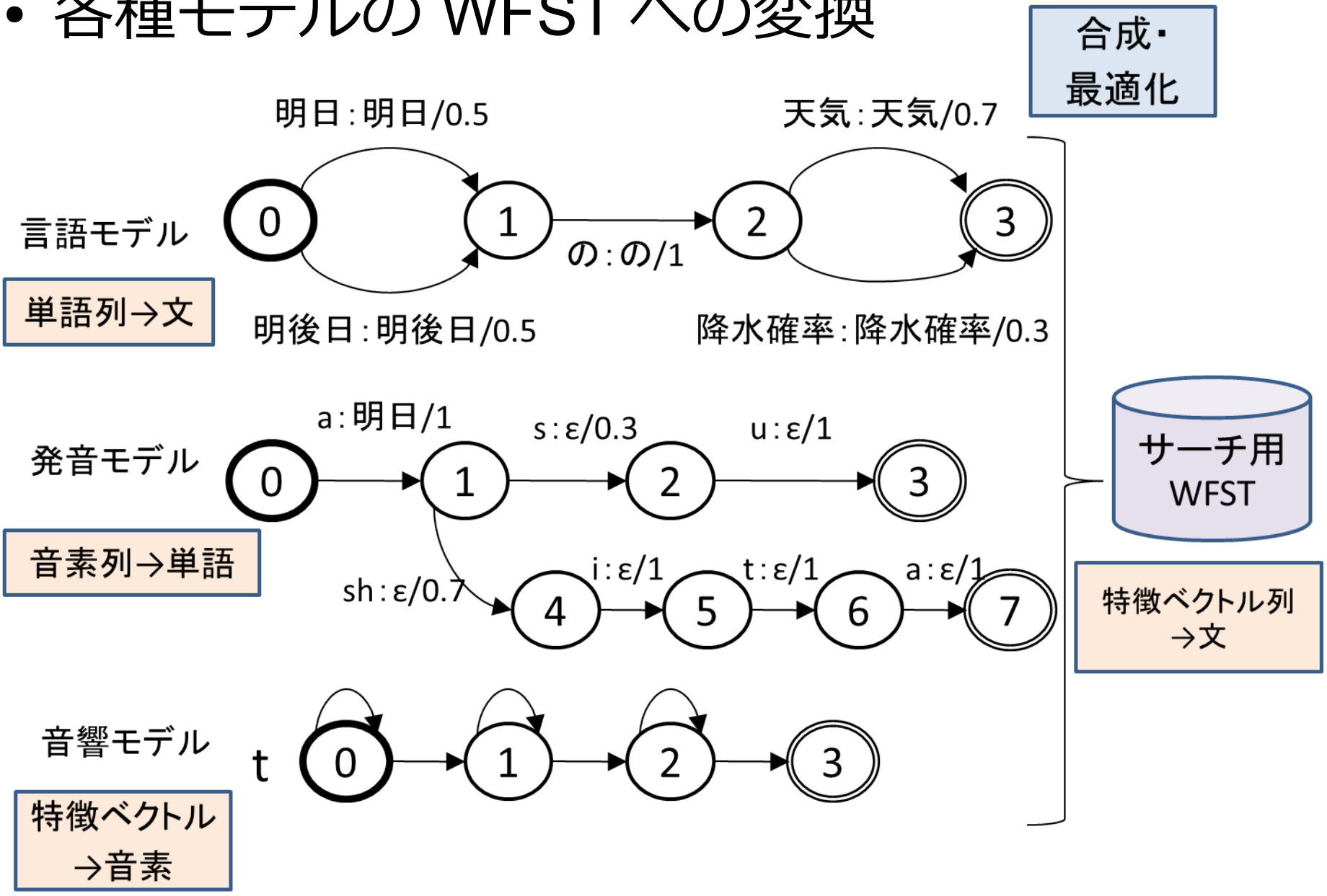
xz+0.0012

14.3 WFST による探索手法

- WFST によるデコードのアイディア
  - 音声認識に用いる確率モデル（ HMM 、単語辞書、言語モデルなど）は WFST で表現可能
  - 記号列 A を記号列 B に変換する WFST1 と、記号列 B を記号列 C に変換する WFST2 を合成すると、記号列 A を記号列 C に変換する WFST になる
    - ただし、状態数は組み合わせ的に増える
  - WFST には、 FSA と同様、決定化・最小化のアルゴリズムが存在する

14.3 WFST による探索手法

- 各種モデルの WFST への変換



14.3 WFST による探索手法

- 利点
  - 探索空間が静的に展開済みなので高速
    - prefix を共有する単語辞書木構造などの効率化が、 WFST の最小化で既に組み込まれている
  - デコードは単純なビームサーチ
    - 言語モデルの適用タイミングなどの工夫が不要
- 欠点
  - 大きなメモリが必要

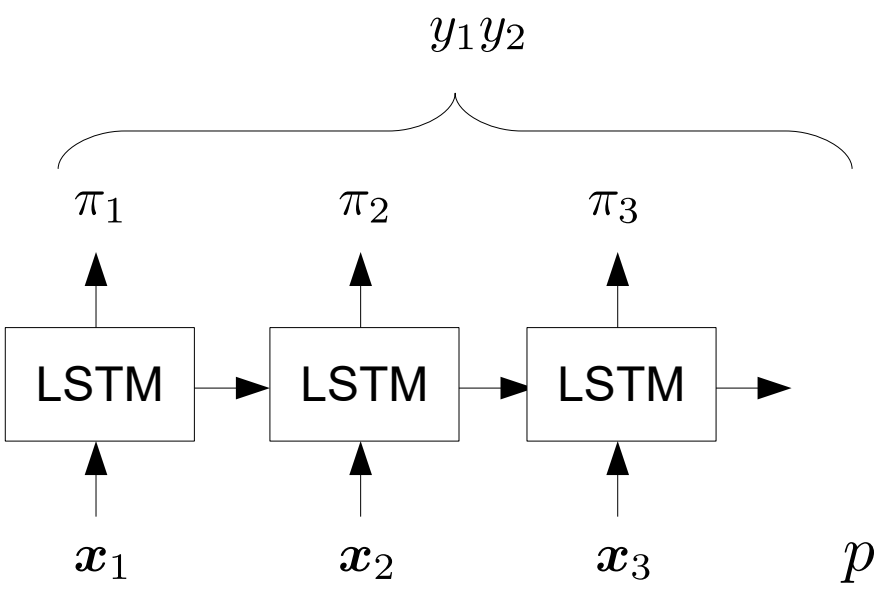
    →探索実行時に合成する on-the-fly 合成も可能

End-to-End アプローチ

- 方法
  - 入力から出力への変換をニューラルネットワークで学習
  - メリット：複雑な工夫が不要
  - 入力：音響特徴量
  - 出力：音素列または単語列
  - ネットワークの構成
    - Connectionist temporal classification
    - Attention モデル

Connectionist temporal classification

- アイディア
  - 出力記号に blank 記号 _ を加えて、入力長と出力長を合わせる
  - 正解系列に変換可能な出力系列の確率の和を求める



- hai という正解系列に対して

  _h___a___i

  hhh_aaaa_i

  hh____ai__

  などの出力系列を正解とみなす

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \to \boldsymbol{y}} p(\boldsymbol{\pi}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \to \boldsymbol{y}} \prod_{i=1}^{T} p(\pi_i|\boldsymbol{x}_i)$$

Attention モデル

- アイディア
  - 特徴ベクトルを分散表現に変換するエンコーダと、分散表現から出力を求めるデコーダの組み合わせ
  - 一定範囲の分散表現から出力を計算するために注意機構 (attention) を用いる