

## 12.4 Julius での文法記述

- Julius とは
  - フリーの音声認識エンジン
  - 有限状態文法 (DFA) に基づいて, 与えられた文法規則の元で入力音声に対して最尤の単語系列を探しだす
  - 統計的言語モデル ( 13 章) も利用可能

## 12.4 Julius での文法記述

- Julius の文法
  - grammar ファイル：構文制約をカテゴリを終端規則として記述する
  - yomi ファイル：カテゴリごとに単語の表記と読みを登録する

## 12.4 Julius での文法記述

- grammar ファイル

#で始まる行は  
コメント

# 文

S: NS\_B KUKAN ZASEKI MAISUU NS\_E

# 区間

KUKAN: EKIMEI KARA EKIMEI MADE

# 駅名

EKIMEI: TIMEI EKI

EKIMEI: TIMEI

# 枚数

MAISUU: SUUJI MAI

NS\_B、NS\_Eはそれぞれ文頭・文末の無音区間

# 12.4 Julius での文法記述

- yomi ファイル

% で始まる行は  
非終端記号

%TIMEI

東京 とーきょー

品川 しながわ

新横浜 しんよこはま

名古屋 なごや

京都 きょーと

新大阪 しんおーさか

%SUUJI

1 いち

2 に

3 さん

%ZASEKI

グリーン席 ぐりーんせき

指定席 してーせき

自由席 じゅーせき

%KARA

から から

%MADE

まで まで

%EKI

駅 えき

%MAI

枚 まい

% NS\_B # 文頭無音

<s> silB

% NS\_E # 文末無音

</s> silE

## 13.2 N- グラム言語モデル

- 3- グラム確率の推定

- 最尤推定を用いる

- $C(w)$ : 単語列  $w$  の出現回数

$$f(w_i | w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}$$

- $P(w_i | w_{i-2}, w_{i-1}) = f(w_i | w_{i-2}, w_{i-1})$  とするとスパースネスの問題が生じる

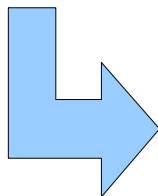
- 妥当な単語列であっても偶然コーパスに出現しなければ3- グラムの確率が 0 になる
- 補間法、スムージングなどで対処

# 13.6 SRILM 入門

## 学習テキスト (.text)

自由席 京都 駅 まで  
グリーン車 名古屋 駅 から 京都 駅 まで  
新大阪 駅 まで 自由席 1 枚  
東京 駅 まで 自由席 1 枚  
新横浜 駅 から 新大阪 駅 まで 指定席  
名古屋 駅 まで 自由席  
...

ngram-count  
コマンド



## back-off trigram (.arpa)

\data\

ngram 1=18

ngram 2=48

ngram 3=99

\1-grams:

-99 <s> -99

-0.9294189 から -7.57829

-0.9294189 まで -7.761997

...

\2-grams:

-0.69897 <s> 京都 -0.9242794

-0.7447275 <s> 新横浜 -0.8808135

...

\3-grams:

-0.4710839 駅 から 京都

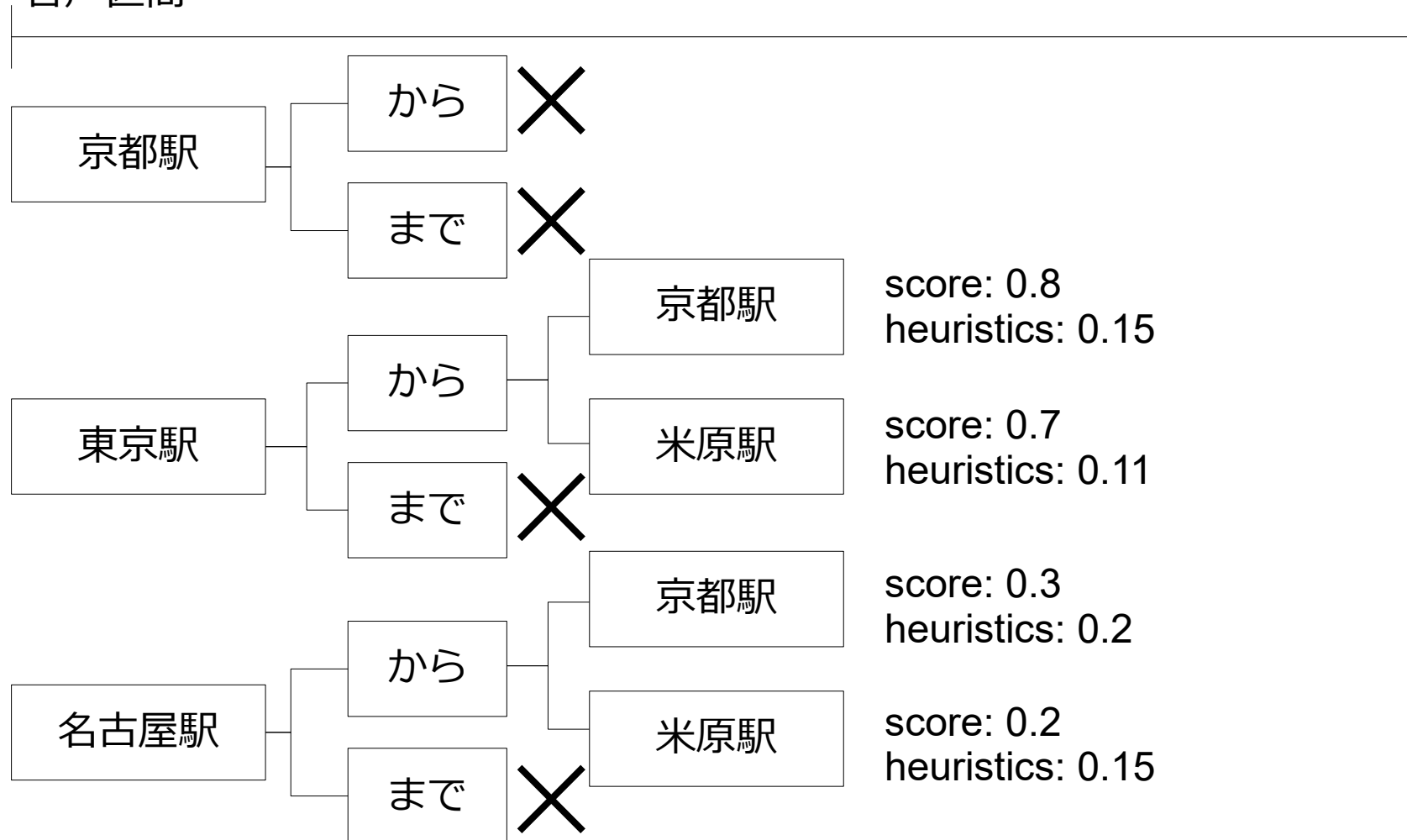
-1.147362 駅 から 新横浜

...

# ヒューリスティックサーチ

- ヒューリスティックサーチとは
  - 各候補の**今後の**スコアを予測し、高い順に探索

音声区間



# ヒューリスティックサーチ

- 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