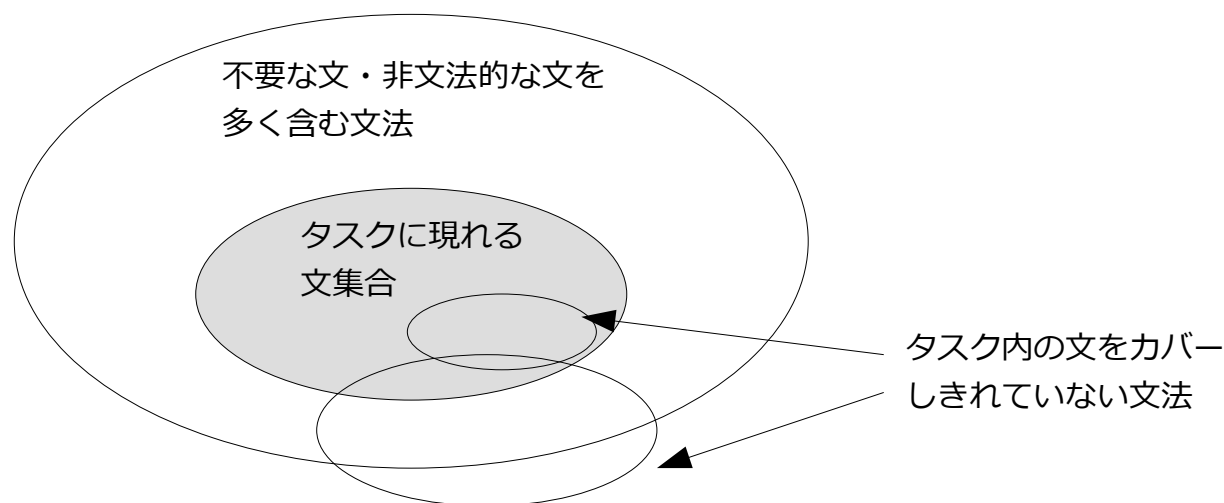


12 章 文法規則を書いてみよう

- 言語モデルとは
 - $P(\text{単語列})$ を計算するための確率モデル
- 2つのアプローチ
 - 文法記述（12章）
 - 単語の並びの規則を人手で記述
 - 使用環境が限定された音声インタフェースなどに適する
 - 統計的言語モデル（13章）
 - 大量のコーパスを元に単語列の出現確率を推定
 - 書き取りシステム（ディクテーション）や用途が限定されない音声認識に適する

12.1 音声認識における文法

- 言語モデルとしての文法
 - 単語から文を構成する規則を文法として記述
 - 文法が受理する単語列 W に対して $P(W) > 0$,
そうでなければ $P(W) = 0$
- 問題点
 - 特定のタスクを対象として入力可能な文を過不足なく文法で定義することは難しい



12.2 タスクから文法を設計する

- 例題タスク
 - 新幹線の切符自動販売機の音声インタフェース
 - 機能
 - 乗車区間を指定できる
 - 席の種類を指定できる
 - 枚数を指定できる
 - 例文
 - 「東京から京都まで自由席 1 枚」
 - 「名古屋から品川までグリーン席 3 枚」

12.2 タスクから文法を設計する

- 文法 = 出現可能な単語列パターンの定義

- 文のパターンを句の並びで定義

– \$ 文 → \$ 区間 \$ 席種 \$ 枚数

例) 東京から京都まで自由席 1 枚

\$ で始まる記号は
非終端記号

- 句のパターンを単語または単語集合の並びで定義

– \$ 区間 → \$ 駅名 から \$ 駅名 まで

– \$ 駅名 → \$ 地名 駅

- 認識対象とする単語集合 (= 語彙) を定義

– \$ 地名 → 東京 | 品川 | 新横浜 | ...

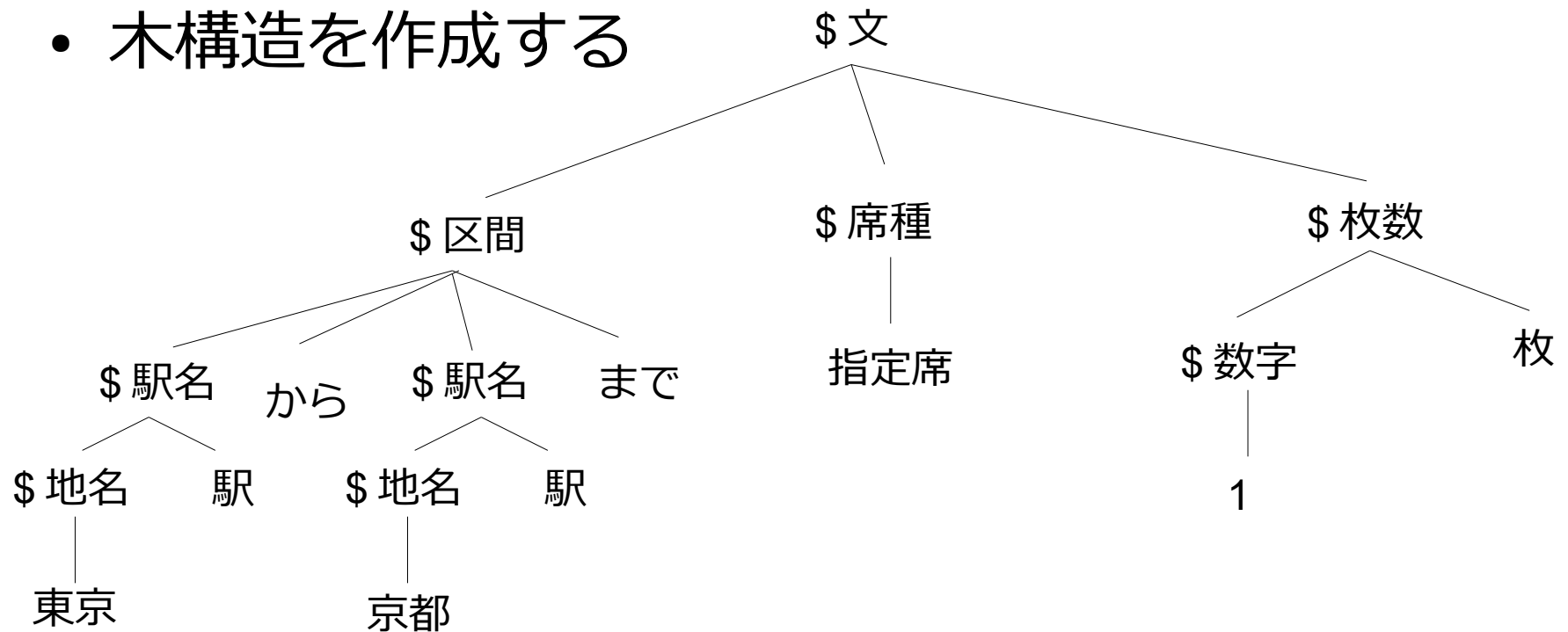
– \$ 席種 → グリーン席 | 指定席 | 自由席

12.3 文法規則における制限

- 記号の集合
 - 非終端記号：文、句、単語集合を表す記号
 - 終端記号：単語
- 規則の書き方
 - 左辺：非終端記号 1 つ
 - 右辺：非終端記号 または 終端記号 の列
- 文法の種類
 - 文脈自由文法：右辺に任意の記号列
 - 正規文法：右辺が終端記号 + 非終端記号
または終端記号

12.3 文法規則における制限

- 文脈自由文法の記述能力
 - 文や句の埋め込みが記述可能
 - 自然言語の大半の部分や、プログラミング言語
- 解析方法
 - 木構造を作成する



12.3 文法規則における制限

- 文脈自由文法の規則

- $A \rightarrow \alpha^+$

A,B,C,... : 非終端記号

a,b, c,... : 終端記号

$\alpha, \beta, \gamma, \dots$: いずれでもよい

- 文脈自由言語の解析方法

- トップダウンパーシング

- 開始記号からスタートし、適用可能な規則を試してゆく

- ボトムアップパーシング

- 単語からスタートし、適用可能な規則を組み合わせてゆく

12.3 文法規則における制限

- 正規文法
 - 正規言語：正規文法で定義された言語
- 正規文法の規則
 - $A \rightarrow a B$
 - $A \rightarrow a$
- 正規文法の解析
 - 正規言語は等価なオートマトンに変換可能
 - 単語数 N に比例した時間で解析可能

文脈自由文法の形式で記述された規則の多くは、正規文法の形式に変換可能

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
 - voca ファイル：カテゴリごとに単語の表記と読み（音素列）を登録する
 - yomi2voca.pl を使えば、ひらがな表記を音素列に変換することができる

12.4 Julius での文法記述

- grammar ファイル

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

- voca ファイル

%TIMEI

東京 t o: k y o:

品川 sh i n a g a w a

新横浜 sh i N y o k o h a m a

名古屋 n a g o y a

京都 k y o: t o

新大阪 sh i N o: s a k a

%SUUJI

1 i c h i

2 n i

3 s a N

%ZASEKI

グリーン席 g u r i: N s e k i

指定席 sh i t e: s e k i

自由席 j i y u: s e k i

%KARA

から k a r a

%MADE

まで m a d e

%EKI

駅 e k i

%MAI

枚 m a i

無音用エントリ

%NS_B # 文頭無音

<s> silB

%NS_E # 文末無音

</s> silE

12.5 標準化された文法記述

- SRGS (speech recognition grammar specification) とは
 - 音声による Web アクセスのための要素技術として W3C が標準化を進めている規格
- SRGS の特徴
 - XML 形式なので機械処理が容易
 - Microsoft Kinect v2 SDK でサポート

12.5 標準化された文法記述

BNF 記法	SRGS
$\alpha \rightarrow \beta \gamma$	<pre><rule id="α"> <ruleref uri="#β" /> <ruleref uri="#γ" /> </rule></pre>
$\alpha \rightarrow a \mid b \mid c$	<pre><rule id="α"> <one-of> <item> a </item> <item> b </item> <item> c </item> </one-of> </rule></pre>
$\alpha \rightarrow a \beta$	<pre><rule id="α"> <item> a </item> <ruleref uri="#β" /> </rule></pre>
$\alpha \rightarrow \beta^*$	<pre><rule id="α"> <item repeat="0-"> <ruleref uri="#β" /> </item> </rule></pre>

12.5 標準化された文法記述

```
<?xml version="1.0" encoding="utf-8"?>
<grammar version="1.0" xml:lang="ja" mode="voice" root="root">
  <rule id="root" scope="public">
    <ruleref uri="#kukan"/>
    <ruleref uri="#zaseki"/>
    <ruleref uri="#maisuu"/>
  </rule>
  <rule id="kukan">
    <ruleref uri="#eki"/>
    <item> から </item>
    <ruleref uri="#eki"/>
    <item> まで </item>
  </rule>
  <rule id="zaseki">
    <one-of>
      <item> グリーン席 </item>
      <item> 指定席 </item>
      <item> 自由席 </item>
    </one-of>
  </rule>
```

...