

## 音声言語のパーズングにおける最適な単語列の探索について

岡田 美智男

NTT 基礎研究所 情報科学研究部

音声言語のパーズングでは、単語列に対して適切な統語構造や意味的な構造を取り出すだけでなく、最適な単語列を効率よく探索するための制御機構としての重要な役割を持っている。本稿では、連続音声から文脈自由文法、およびより一般的な句構造文法などの制約の下で、適切な単語列とその単語列に対する意味解釈を取り出すための基本的なアルゴリズムを示す。提案する手法は、ワードラティスやフレーズラティスなどを中間表現として用いることなく、音声のフレームレベルの特徴系列から直接に文の候補やその意味解釈を取り出すことができる。そのため、階層間での情報の受け渡しにおけるオーバーヘッドを回避でき、談話レベルから得られる様々な制約を直接、音声のフレームレベルでの探索に反映させることが可能である。また、音声言語のパーズングにおける探索と解釈過程の役割について考察した。

### On an Optimal Word Searching in Spoken Language Parsing

Michio OKADA

*Information Science Research Laboratory,  
NTT Basic Research Laboratories  
3-9-11, Midori-Cho Musashino-Shi, Tokyo, 180, Japan  
( E-mail: okada@atom.NTT.jp )*

We have been investigating the basic architecture of a spoken language system generally used for dialogue speech understanding. In particular, our main interest is in the use of predictive information about the utterance goals and plans, which is partly inferred by a formal planning system.

We discussed in this paper how to handle sub-sentence alternatives and structural ambiguities for generating the minimal network and for one-pass searching. We used the augmented active chart parsing for the network generation, which uses a bookkeeping method known as a well-formed substring table. The architecture proposed here can reduce search spaces by using dynamically varying constraints, as well as grammatical constraints.

# 1 はじめに

連続音声から、最適な単語列だけでなく、適切な意味解釈を取り出すための汎用的な音声言語システム (Spoken Language System) の研究を進めている。音声言語システムの研究では、(1) 音声処理と言語処理のインターフェースをどのように考えていくか、(2) spontaneous な音声現象をどのように扱うか、(3) プロソディ等の音声固有の特徴を文の解析や談話処理のレベルにどのように反映させていくか、(4) 言語モデル、談話モデルとして、どのような枠組を用意するか、そしてそれをどのように音声レベルのサーチに反映させていくか、等の興味ある問題が当面の研究課題として考えられている。

本稿は、主に、(1)の問題に対して検討を行なったもので、連続音声から文脈自由文法、およびより一般的な句構造文法などの制約の下で、適切な単語列とその単語列に対する意味解釈を取り出すための基本的なアルゴリズムを示す。

ここで提案する手法は、従来の階層型のシステムで用いられていたワードラティスやフレーズラティスなどを中間表現として用いることなく、音声のフレームレベルの特徴系列から直接に文の候補やその意味解釈を取り出すことができる。そのため、階層間での情報の受け渡しにおけるオーバーヘッドを回避でき、談話レベルから得られる様々な制約を直接、音声のフレームレベルでの探索に反映させること可能である。同様の観点から、Ney [1] は C.K.Y. 法に基づく連続音声認識アルゴリズムを示している。

この Ney のアルゴリズムは、(1) 基本的に音声の分析フレーム数 (フレームのシフト幅を 5msec 程度とし、5 秒の連続音声进行分析したとすると、フレーム数は約 1000 フレームとなる) の 3 乗に比例したオーダの計算量を必要とする、(2) 制御構造が基本的にボトムアップであり、すべての語彙や統語規則に対して網羅的な探索を行なう、などの問題点が指摘されている。

ここで提案するアルゴリズムは、One-Pass サーチ [2]、Viterbi サーチなどの音声のフレームレベルでの探索アルゴリズムを直接、文脈自由文法などのより一般的な制約の下で制御する枠組であり、音声のフレーム数に対してリニアオーダの処理となる。また、フレームに同期した探索を行なうなどの優れた性質を持っている。

以下、本稿では、(1) 現在、NTT 基礎研究所で構築を進めている音声言語システムの概要、(2) One-Pass サーチアルゴリズムを文脈自由文法で制御するための基本的な拡張、(3) One-Pass サーチアルゴリズムを動的に制御するための、チャート解析法に基づいたミニマルなネットワークの生成法、等について述べ、最後に、音声言語パーズングにおける探索と解釈の二つの過程、すなわち連続音声から最適な単語列を探索する過程と、単語列に対して統語的な構造や意味的な解釈を取り出す過程、相互の分離可能性について考察する。

# 2 音声言語システムの概要

ここ数年、米国の DARPA の音声認識に関する研究プロジェクトのターゲットが、不特定話者を対象とした大語彙連続音声認識の研究から、「音声言語システム

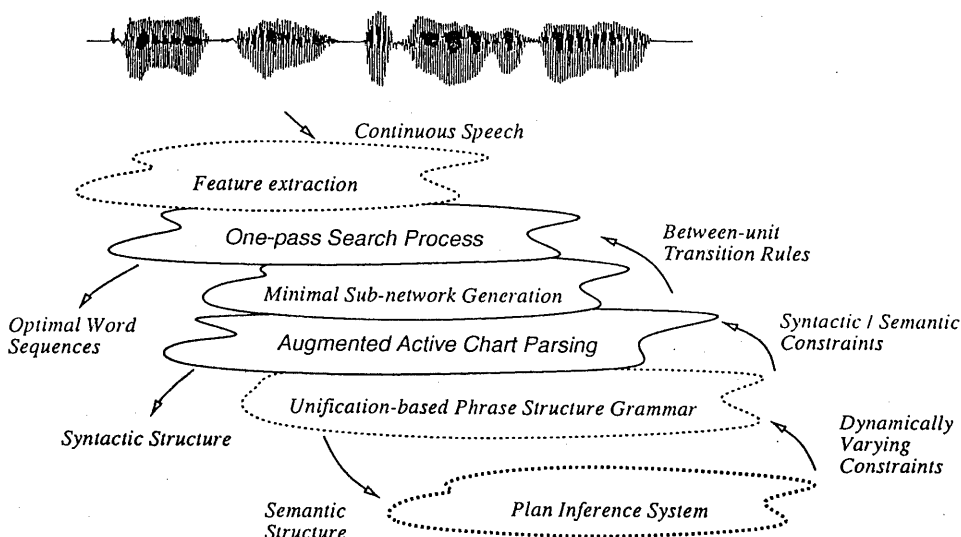


Fig. 1 音声言語システムの概要

(Spoken Language System)」と呼ばれる、音声と自然言語処理とを“密に”統合したシステムの研究に移行しており、その動向が注目される [3]。これらのシステムでは、従来の Dictation Machine などを意図した大語彙連続音声認識のように、連続音声から最適な単語列を取り出すだけでなく、その単語列に適切な意味解釈を与えたり、談話レベルの制約を音声の探索に直接反映させるための基本的な枠組を必要とする。特に、spontaneous な音声を扱うためには、より能動的な枠組を必要とし、発話の生成過程に関する深い理解が不可欠である。例えば、spontaneous な音声で発話をする際のプランニングのモードは、written language のそれとは異なり、この違いが ill-formed な発話文を解析する際のポイントになると考えられる。

このような観点から、Fig.1 に示すような音声言語システムの構築を進めている。対話の中で、相手の発話を理解する際には、話し手の発話意図に関する断片的な情報やその発話がなされる状況に関する知識が重要な役割を果たす。研究の焦点の一つは、対話を進めていく過程で、話し手の発話意図が断片的にでも推論された際に、これらの部分的な情報が発話を効率的に解析する上で、どの程度寄与するのか、という点である。そのため、静的な制約だけでなく、プラン推論等の過程で得られる、動的で、断片的な制約を音声のサーチに直接反映させる枠組が重要である。

ここでは、素性構造の単一化に基づくバージョンアーキテクチャと One-Pass サーチアルゴリズムとを統合することにより、音声の探索と言語処理の探索アルゴリズムが“密な”結合関係にある統合アーキテクチャを実現している。素性構造の単一化の機構は、形態レベル、統語レベル、談話レベルやプロソディ等の様々な制約を宣言的、かつ統一的に扱うことができる。また、断片的な情報の扱い、制約を bi-directional に伝搬させるための機構、意味解釈を取り出すための機構など、幾つかの優れた性質を兼ね備えている。これは、次発話の予測に関する断片的な情報や動的に変化する様々な制約を扱うための基本的な機能を提供する。

また、様々な制約を音声のフレームレベルでのサーチアルゴリズムに反映させるために、ここでは拡張されたアクティブチャートパーザが中心的な役割を果たしている。これは、より一般的な制約から、発話の解析と同時に One-Pass サーチのプロセスを制御するためのネットワークをインクリメンタルに供給していく汎用的な枠組になっている。システムのトータルなパフォーマンス(認識精度および効率性)を向上させるためには、様々な制約によって探索空間を最小化し、それを反映したミニマルなネットワークを生成することが重要なカギとなる。

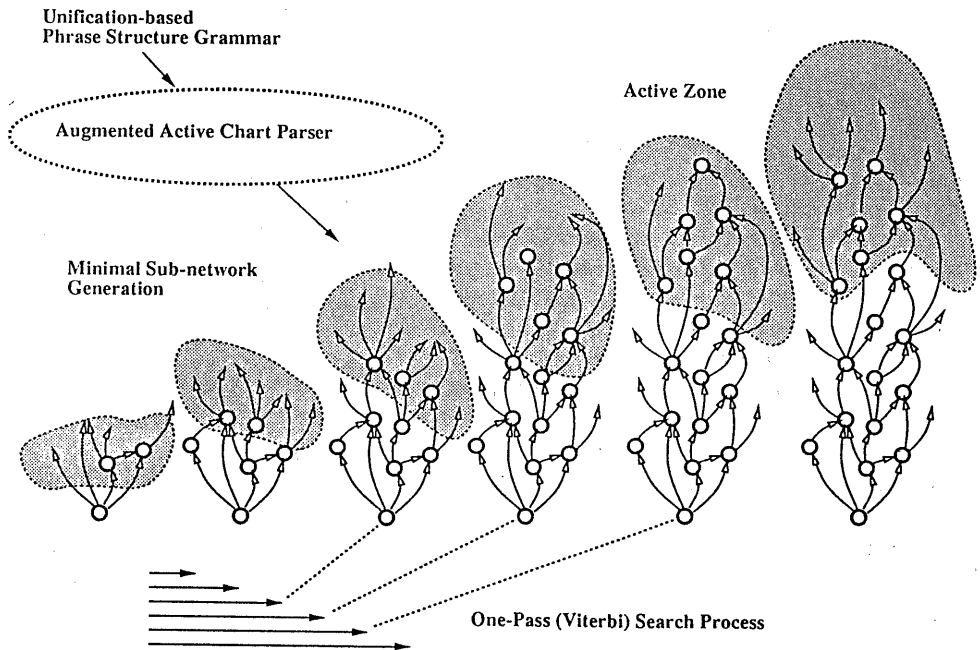


Fig.2 One-Pass サーチの動的な制御

### 3 単語列の探索アルゴリズム

音声処理と言語処理とのインターフェースの考え方として、単語や句などのラティスなどを中間的な表現として用いるラティスパージングのアプローチがある[4-5]。また、日本語を対象とした連続音声認識システムでは、文節内と文節間で文法の表現方法や処理アルゴリズムを積極的に分離しようとするものが多い[6]。これらの方法の問題点は、(1) 単語列としての最適性が十分に保証されない、(2) 言語的な制約が単語列の認識、サーチのレベルに反映されにくい、(3) 同様な処理を階層間で重複して行なう、などの幾つかのオーバーヘッドの存在である。

ここでは、単語列の最適性が保証されている連続単語認識アルゴリズム(例えば、One-Pass サーチ法[2])を文脈自由な句構造文法、および単一化文法などのより一般的な制約に基づいて制御する方法を示す。この方法では、One-Pass サーチ法を制御するネットワークについて、音響的な整合度の高いアクティブな部分(ここでは、アクティブゾーンと呼ぶ)だけに焦点をあてたサーチを行なう。これは、ネットワークの各エッジそのものに対する枝刈りとは本質的に異なり、ネットワーク中のエッジは決して削除されない。ネットワーク上の焦点をあてる部分、すなわちアクティブゾーンをフレーム同期型のビームサーチの手法[7]を用いて、データ駆動的に遷移させていく。Fig.2にOne-Pass サーチアルゴリズムにおけるネットワークとそのアクティブゾーンの関係を示す。入力音声のフレームに同期してアクティブゾーンを更新し、音声に適応させながら、必要な部分のネットワークをインクリメンタルに追加する。次に示すアルゴリズムは、One-Pass サーチアルゴリズムを、チャート解析法によって生成されたネットワークにより制御するための手続きを示したものである。

One-Pass サーチプロセスでは、入力音声のフレームに同期して、チャートの中の特に語彙エッジから構成される遷移ネットワークにガイドされながら、探索を進める。ここで、 $D_i(j, k, v)$  および  $B_i(j, k, v)$  は、ネットワークの頂点  $v$  から分岐する  $k$  番目の単語(あるいは音節、音素等の適当な単位)の  $j$  フレームに対応する格子点における、累積尤度とバックポインタである。また、 $D(j, k, v)$  および  $B(j, k, v)$  は直前のフレームでの値を保持するためのテンポラリなバッファである。各入力音声のフレーム  $i$  において、ネットワーク上のアクティブゾーン( $AZ_s(k, v) \sim AZ_e(k, v)$ )での累積尤度を計算し、関数  $Active\_zone$  は、その値から次フレームでのアクティブゾーンを推定する。もし、活性な語彙エッジの終端にすでにアクティブゾーンが到達していれば、関数  $Create\_passive\_edge$  は、チャート解析のプロセスに対して、新たな非活性の語彙エッジとして提案する。

この提案は、非同期で行なわれる。一方、チャート解析のプロセス( $Update\_chart$ )では、あらたに提案されたエッジに基づいて、完了操作、予測操作を行ない、チャートの内容を更新する。次のフレームでの処理は、この新たに更新されたチャートを参照して行なう。

#### Initialization of One-Pass algorithm:

```
Chart ← Make_chart_system(Start_symbol);
for each outgoing lexical edge  $LE_{k,1}$  at initial vertex do
   $w_{k,1} \leftarrow Lexical\_item(LE_{k,1});$ 
  for each frames  $j$  of  $w_{k,1}$  do
     $D(j, k, 1) \leftarrow Initial\_cumulative\_score(j, k, 1);$ 
     $B(j, k, 1) \leftarrow 0;$  /* Backpointers */
```

#### Frame-synchronous Process :

```
for input speech frame  $i = 1$  to  $N$  do
  for each vertex  $v \in SubNetwork$  do
    for each outgoing lexical edge  $LE_{k,v}$  do
       $w_{k,v} \leftarrow Lexical\_item(LE_{k,v});$ 
      ( $AZ_s(k, v), AZ_e(k, v)$ ) ←  $Active\_zone(LE_{k,v});$ 
      for  $j = AZ_s(k, v)$  to  $AZ_e(k, v)$  do
        if start_frame?( $j, w_{k,v}$ )
          then /* between-unit transition rule */
             $k' \leftarrow \{incoming\ lexical\ edges\ at\ v\};$ 
             $D_i(j, k, v) \leftarrow max_{k'} \{D(L(k', v), k', v)\};$ 
             $B_i(j, k, v) \leftarrow i - 1;$ 
          else /* within-unit transition rule */
             $j^* \leftarrow argmax \{D(j - 1, k, v); D(j - 2, k, v)\};$ 
             $D_i(j, k, v) \leftarrow D(j^*, k, v) + d(i, j, k, v);$ 
             $B_i(j, k, v) \leftarrow B(j^*, k, v);$ 
        {end of loop}
        if active_edge?( $LE_{k,v}$ )
          and  $AZ_e(k, v) = L(k, v)$  then
            Edge ←  $Create\_passive\_edge(LE_{k,v}, Chart);$ 
            Add_item_to_agenda(Edge, Agenda);
        {end of loop}
         $k \leftarrow \{incoming\ lexical\ edges\ at\ v\};$ 
         $U(i, v) \leftarrow argmax_k \{D_i(L(k, v), k, v)\};$  /*unit*/
         $V(i, v) \leftarrow start\ vertex\ of\ U(i, v);$  /*vertex*/
         $F(i, v) \leftarrow B_i(L(U(i, v), v), U(i, v), v);$  /*frame*/
        SubNetwork ←  $Update\_chart(Chart);$ 
      {end of loop}
    for all  $j, k, v$  do
       $D(j, k, v) \leftarrow D_i(j, k, v);$   $B(j, k, v) \leftarrow B_i(j, k, v);$ 
    {end of loop}
```

#### Traceback Process:

```
Solutions ←  $Trace\_back(Chart, U, V, F);$ 
```

$U(i, v)$ 、 $V(i, v)$ 、および  $F(i, v)$  は、サーチが終了した時点で、最適な単語列をトレースバックして探し出すために用いられるポインタであり、詳細は省略する。単位内の遷移ルール(within-unit transition rule)は、通常の時間軸の非線形整合を行なうための傾斜制限であり、One-Pass サーチアルゴリズムでは単位間の遷移ルール(between-unit transition rule)が中心的な役割

を果たす。このルールは、ネットワーク中に合流点がないと全く機能しないことに注意する必要がある。

個々の単語の標準パターン(あるいは、単語モデル)は、音韻環境に依存したセグメントを連結することにより構成するため、語彙記述の変更によって基本的には任意の語彙を扱うことができる。ここでは、DP に基づいた定式化を行ないアルゴリズムの概要を示したが、HMM では、Viterbi サーチを用いることでアルゴリズム、枠組に大きな変更はない。このアルゴリズムでは、単位間の遷移ルールを示すネットワークをインクリメンタルに展開していくための枠組が必要になる。つぎに、最小のネットワークを用意するためのチャート解析法を示す。

#### 4 音声言語のパーズング

入力音声の発話内容に適応させてミニマルなネットワークを生成するためには、サーチアルゴリズム側からのデータ駆動的な情報に加え、チャートの制御方略が重要な要素となる。チャート解析法 [8] は、柔軟で効率のよい汎用的なパーズングの枠組として自然言語処理の分野で広く利用されている。音声言語に対するパーズングでも、不適格 (ill-formed) な入力や曖昧性の高い入力から、発話の内容を解析するためにチャートのデータ構造や制御方略が幾つか検討されてきた [5][9-10]。

Thompson [9] は、ラティスで表現された単語系列から、動的計画法の方略を用いて、スコアの高い(いわゆる、N-best な)単語系列を捜し出す汎用性の高いチャート解析法を提案している。本稿で述べるチャート解析法は、この Thompson の枠組を、音声のフレームレベルに拡張したものと考えることができる。すなわち、ラティスなどの中間的な表現を介さず、直接フレームレベルから最適な単語系列とその統語構造を取り出す。また、談話レベルから得られる様々な制約をフレームレベルのサーチ (One-Pass サーチ) に反映させることができる枠組になっている。

##### 4.1 ミニマルなネットワークを与えるアクティブチャート解析法

チャート解析法は、解析の部分結果を表 (WFST) を用いて管理し、重複のない効率的な解析をする。(a) 解析開始の段階で、あらかじめ単語の候補がデータ駆動的に用意されていない、(b) 複数の可能性 (alternatives) を並行して調べ、相対的な比較操作により、最適パスを探索していく、という音声言語のパーズングの性質を考慮すると、基本的に下降型で横型の解析方略となる。アルゴリズムの概要を以下に示す。チャートの初期化では、アジェンダ (Agenda) の中身をクリアし、開始記号が  $S$  で始まるルールに相当するエッジをアジェンダに加える。

##### Initialization of Chart:

```
Clear_agenda(Agenda);
for each rule  $S \Rightarrow \alpha_1\alpha_2\dots\alpha_n$  do
  Edge  $\leftarrow [S \Rightarrow \alpha_1\alpha_2\dots\alpha_n, 0, 0, \{\}];$ 
  Add_item_to_agenda(Edge, Agenda);
{end of loop}
```

ここで、各エッジを次のような 4 つ組によって表現する。

$$edge = [Dotted-rule, i, j, \{n, m, l, \dots\}]$$

ここで、 $i, j$  はそれぞれ start vertex、end vertex である。また、 $\{n, m, l, \dots\}$  はオルタナティブである娘のエッジに対してリンクを張るためのポインタのリストを表す。

##### Main Procedure for Updating Chart:

```
while not Agenda = nil do
  Edge  $\leftarrow$  Get_item_from_agenda(Agenda);
  case EdgeSource_Type of
    Prediction : /* for predicted edge */
      if not Subsuming?(Edge, Chart) then
        Add_item_to_chart(Edge, Chart);
        Prediction_and_completion(Edge);
    Completion : /* for completed edge */
      EqualEdge
         $\leftarrow$  Search.EqualParentEdge (Edge, Chart);
      if EqualEdge = nil then
        Add_item_to_chart(Edge, Chart);
        Prediction_and_completion(Edge);
      else
        Append_pointer_list(Edge, EqualEdge);
    Insertion : /* for inserted lexical edge */
      Add_item_to_chart(Edge, Chart);
      Prediction_and_completion(Edge);
  {end of case}
{end of while}
```

チャート解析は、エッジをアジェンダから選択し、そのエッジを表 (Chart) に加え、予測操作 (prediction step) と完了操作 (completion step) と呼ばれる二つの手続きを進め、その結果得られたエッジを再度アジェンダに加えることを繰り返して行なう。アジェンダから取り出されたエッジを表に加える際に、重複したエッジを加えることを避けるためのチェック (Subsuming?) や、オルタナティブの検出 (Search.EqualParentEdge) が行なわれる。すなわち、完了操作の際に、すでに同様な機能を持った親のエッジが存在すれば、同様な親エッジを新たに加えることは避け、すでに存在する親エッジに、ポインタによってオルタナティブな娘エッジとして帰属させる。この、ポインタリストのアペンドの操作は、Append\_pointer\_list により行なわれる。

One-Pass サーチアルゴリズムでは、動的計画法の性格から探索の途中段階で大局的な意味で最適な単語や

その境界を決定することができない。従って、ここで設定される頂点は時間軸から独立したものとなる。そのため、Thompson のチャートのように探索の過程で生じるオルタナティブを同一の頂点に合流させることはできない。一方、回帰ルールによって生じる時間軸方向でのオルタナティブも通常のオルタナティブと同様に扱え、ミニマルなチャートを構成することができる。

Procedure Prediction\_and\_completion(Edge):

```

case Edge of
  [A ⇒ α · Bβ, i, j, {o, p, ...}] : /* for active edge */
  /* prediction step */
  for each rule B ⇒ γ1...γn do
    Edge ← [B ⇒ γ1...γn, j, j, {}];
    Add_item_to_agenda(Edge, Agenda);
  /* completion step */
  for each [B ⇒ γ, j, k, {m, n, ...}] on Chart do
    l ← Create_new_vertex(Chart);
    Edge ← [A ⇒ αB · β, i, l, {k}];
    Add_item_to_agenda(Edge, Agenda);

  [B ⇒ α, i, j, {o, p, ...}] : /* for passive edge */
  /* completion step */
  for each [A ⇒ β · Bγ, k, i, {m, n, ...}] on Chart do
    l ← Create_new_vertex(Chart);
    Edge ← [A ⇒ βB · γ, k, l, {j}];
    Add_item_to_agenda(Edge, Agenda);
{end of case}

```

また、同一性のチェックは以下の手続きによる。

Procedure Search.EqualParentEdge (Edge, Chart):

```

case Edge of
  [A ⇒ α · BC, i, j, {o, p, ...}] : /* for active edge */
  if ExistOnChart? ([A ⇒ β · BC, i, k, {q, r, ...}]) then
    return ([A ⇒ β · BC, i, k, {q, r, ...}]);
  else return(nil);

  [B ⇒ α, i, j, {o, p, ...}] : /* for passive edge */
  if ExistOnChart? ([B ⇒ β, i, k, {q, r, ...}]) then
    return ([B ⇒ β, i, k, {q, r, ...}]);
  else return(nil);
{end of case}

```

4.2 動作例

Fig.3 に、最も一般的なオルタナティブに対するチャートのデータ構造と、それから得られる遷移ルールを表現する仮想的なネットワークの例を示す。ある名詞句 (NP) が頂点 {4} から予測されているものとし、そこで展開されるチャートの構造とそれに対応する遷移ネットワークを示している。この解析過程をトレースすると次のようになる。

- (1): [NP → .AN, 4, 4, {}]
- (2): [A → .“aoi”, 4, 4, {}] ... predicted from (1).

- (3): [A → .“akai”, 4, 4, {}] ... predicted from (1).
- (4): [NP → .N, 4, 4, {}]
- (5): [N → .“hana”, 4, 4, {}] ... predicted from (4).
- (6): [N → .“hako”, 4, 4, {}] ... predicted from (4).

(1)～(6) までのステップは、通常のチャート解析法における予測操作と同様であり、この時点で4つの語彙エッジが予測される。

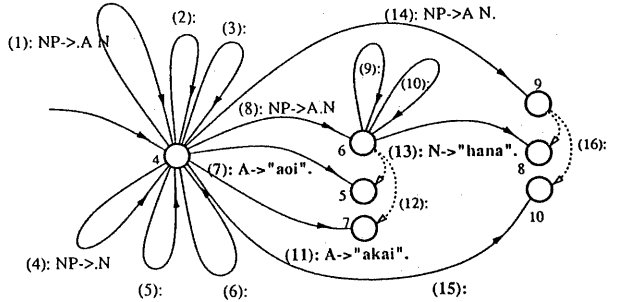
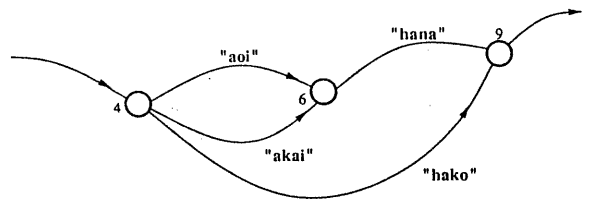


Chart data structure for the sub-sentence alternatives.



Corresponding sub-network structure

Fig.3 チャートのデータ構造と動作例

One-Pass サーチのプロセス (OP) では、サーチの進行に伴いアクティブゾーンを更新する。もし、このアクティブゾーンが語彙エッジの終端に到達した場合、この語彙エッジに基づいてチャートに対して非活性の語彙エッジを提案する。

- (7): [A → “aoi”., 4, 5, {}] ... proposed from OP.
- (8): [NP → AN, 4, 6, {5}]
  - ... combining (7) with (1).
- (9): [N → “hana”, 6, 6, {}] ... predicted from (8).
- (10): [N → “hako”, 6, 6, {}] ... predicted from (8).
- (11): [A → “akai”., 4, 7, {}] ... proposed from OP.
- (12): Modifying (8)
  - to [NP → AN, 4, 6, {5, 7}] from (11).
- (13): [N → “hana”., 6, 8, {}] ... proposed from OP.
- (14): [NP → AN., 4, 9, {8}]
  - ... combining (13) with (8).
- (15): [N → “hako”., 4, 10, {}] ... proposed from OP.
- (16): Modifying (14)

to  $[NP \rightarrow AN., 4, 9, \{8, 10\}]$  from (15).

(7), (11), (13), (15) などが、One-Pass サーチのプロセスから提案されたエッジであり、このエッジに基づいて完了操作が行なわれる。完了操作において、新たに加えられる親エッジが、その要因 (source) となった娘エッジに対するリンク情報を保持する。(12), (16) のステップでは、オルタナティブな娘エッジを共有化している。すなわち、すでに同様な機能を持った親のエッジが存在すれば、同様な親エッジを新たに加えることは避け、すでに存在する親エッジに、オルタナティブな娘エッジとして新たに加えておく。

これらのリンク情報に基づいて、語彙エッジを結びつけていくと、対応するネットワークが構成される。

## 5 インプリメンテーション

上述した枠組に基づく音声言語システムを構築した。このシステムは文節境界等をとくに意識せず連続に発声された音声を対象とする。素性構造の単一化の枠組で記述された制約の下で、最適な単語系列とその統語的、意味的な構造を抽出する。汎用的な枠組であり、特定のタスクに適用する際には、文法記述や語彙記述の枠組を変更するだけでよい。

システムは、最適な単語列を探索する One-Pass サーチプロセスと、この単語列のサーチに対する統語的な制約を与えるための遷移ネットワークをインクリメンタルに生成する、チャート解析のプロセスとが非同期的に接続された構成となる。プロセス間での通信量は比較的小さい。インプリメントを行なう際には、これらのプロセスを幾つかの計算機に機能分散させることが考えられが、この時、プロセス間の通信によるオーバーヘッドは問題にならないほど小さい。処理時間等は、入力される連続音声や文法記述の内容によって大きく影響を受けるが、現在の実験の規模では、1 文章の処理時間は数秒のオーダーである。

## 6 考察

### 6.1 インターフェースの考え方

ここでは、音声のフレームレベルでのサーチプロセスに対して、様々な言語的制約を反映した単語の遷移ネットワークをインクリメンタルに供給するという観点から、音声言語のパーズングの枠組をとらえた。また、サーチプロセスに対して「必要最小のネットワーク (minimal network)」を用意するという観点で、最適な単語系列のサーチとパーズング処理の効率の問題をとらえた。

このような“密な”統合関係にあるアーキテクチャ

では、階層間でオーバーヘッドが回避でき、最適な解を得ることができる。反面、システム全体の処理が重くなりやすいという傾向がある。この問題を回避するために、大語彙を対象としたシステムでは、一般的な句構造文法などを有限状態ネットワークで近似する [11-12]、あるいは言語的な制約を二つに分離し、単語列の探索では比較的緩い制約を用いることが多い [13-14]。しかし、これらのアプローチでは、いわゆる言語的制約がフィルタリングとしてしか機能せず、談話レベルからの予測情報が有効に利用できない。ここでは、予測的な情報が反映できる枠組を残したまま、最適な単語列の探索とパーズングにおける解釈の過程を意識的に分離した枠組となっている。

### 6.2 探索と解釈フェーズの分離

連続音声をフレームレベルでパーズする際には、性質の異なる二種類の“曖昧さ”が存在する。すなわち、(1) 統語規則中に曖昧さが含まれる場合に、ある局所的な単語系列について、構造的に複数の解釈が可能な場合がある。この曖昧さは、他の談話的な情報や音声的な情報が付加されない限り、一意化されない。(2) 音声認識の不確実さを考慮すると、単語系列として幾つかの候補、あるいは可能性 (sub-sentence alternatives) を保持しながら探索していく必要がある。これらのオルタナティブは、One-Pass サーチにより最終的には最適な単語系列として一意化されるものであるが、探索の途中段階では最適な単語系列は知り得ない。

これらの曖昧さの性質を考慮して、パーズングの処理の中から探索フェーズだけを分離することを考える。Fig.4 にその概念図を示す。

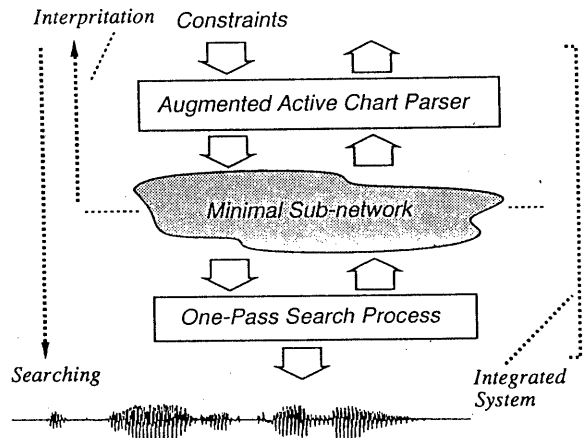


Fig.4 探索と解釈フェーズの関係

ここでは、インクリメンタルに生成されるミニマルなネットワークを介して、情報の流れを制御する。基本的には、チャート解析法により、各レベルで次のような方略が可能である。

1. 音声の探索におけるオルタナティブ (音声レベルでの曖昧性) の影響を可能な限り解釈レベルに反映させない。

オルタナティブのパス (単語の部分系列) に対して、独立に部分木を与えると、部分木の総数は、可能なパスの組合せ数 (指数オーダ) に膨れ上がる。

2. パージングの過程で生じる構造的な曖昧性の影響を単語列の探索レベルに反映させない。

ある局所的な単語系列について、構造的に複数の解釈が可能な場合がある。探索の際には、曖昧さを含んだ構造を可能な限りバックし、以降の解析に曖昧さをひきずらない。この探索が終了した時点で、曖昧な構造を再構築する枠組を用意することができる。

3. 素性構造のなかで、探索に関与せず、解釈結果だけを親に伝搬させる素性の具現化を遅らせる。

単語列の探索に部分的な意味解釈の結果を必要としないならば、膨大なオルタナティブに対する個々の意味解釈等は、この時点では必要でない。最適な単語系列が決定した後、解析木の作成や、バックキングされた部分木の解凍、意味の解釈を行なってもよい。解釈の過程では、探索の過程で作られたチャートをそのまま利用することができる。

統合化された様々な制約に対して、確率的な重み付けを行なうことは難しいが、一度ネットワークにブレークダウンされるとネットワークの遷移確率として、幾つかの情報を付与することが容易になる。また、我々が無意識に発話の解析を行なう際には、低いレベルでネットワークサーチのようなニアオーダの解析を行なっている可能性がある。高次レベルでの処理との切り分けの問題など、包摂アーキテクチャにも関連した興味ある問題がまだ残されている。

## 7 おわりに

音声のフレームレベルでのサーチアルゴリズムと統語処理におけるパージングアルゴリズムとを適切に融合する手法を示した。特に、チャート解析の枠組に基づいてミニマルなネットワークを動的に生成するための、具体的なチャートのデータ構造と制御方略を示した。チャート解析法は、LR法と異なり動的な予測を行なう。そのため、発話の内容、対話の状況によって制約を動的に変更するような柔軟なパージングが行なえる。今後は、よ

り自然な発話 (spontaneous speech) を扱うための制御方略を考えていきたい。

## 謝辞

日頃、御討論いただく NTT 基礎研究所 島津明 主幹員、竹内研究グループの諸氏、計量計画研究所 大塚 裕子 研究員に感謝いたします。

## 参考文献

- [1] H.Ney : "Dynamic Programming Speech Recognition using a Context-free Grammar", Proc. ICASSP-87, pp.69-72 (1987).
- [2] J.S.Bridle, M.D.Brown and R.M.Chamberlain : "An Algorithm for Connected Word Recognition", Proc. ICASSP-82, pp.899-902 (1982).
- [3] 例えば、Proceedings of the DARPA Speech and Natural Language Workshop, Morgan Kaufmann (1989-02).
- [4] M.Tomita : "An efficient word lattice parsing algorithm for continuous speech recognition", Proc. ICASSP-86, pp.1569-1572 (1986).
- [5] L.Chien, K.J.Chen and L.Lee : "An Augmented Chart Parsing Algorithm integrating Unification Grammar and Markov Language Model for Continuous Speech Recognition", Proc. ICASSP-90, pp.585-588 (1990).
- [6] 岡田, 松尾, 伊藤, 小川, 牧野, 城戸 : "文節スポッティングに基づく日本語文音声の認識", 電気学会論文誌 (C), Vol.108-C, No.10, pp.826-833 (1988-10).
- [7] H.Ney, D.Mergel, A.Noll and A.Paeseler : "Data-Driven Organization of a Dynamic Programming Beam Search for Continuous Speech Recognition", Proc. ICASSP-87, pp.833-836 (1987).
- [8] M.Kay : "Algorithm Schemata and Data Structures in Syntactic Processing", Technical Report CSL-80-12, Xerox PARC (1980).
- [9] H. Thompson : "Best-first enumeration of paths through a lattice - an active chart parsing solution", Computer Speech and Language, 4, pp.263-274 (1990).
- [10] G.Satta and O.Stock : "Formal Properties and Implementation of Bidirectional Charts", Proc. AAAI - 90, pp.1480-1485 (1990).
- [11] F. Pereira and R. Wright : "Finite-state Approximation of Phrase Structure Grammars", Proc. of the 29th ACL, pp.246-255 (1991).
- [12] A. Black : "Finite-state Machines from Feature Grammars", International Workshop on Parsing Technologies, pp.277-285 (1989).
- [13] R. Schwartz and Y-L. Chow : "The N-Best Algorithm: An Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses", Proc. ICASSP-90, pp.81-84 (1990).
- [14] R. Pieraccini and C-II. Lee : "Factorization of Language constraints in Speech Recognition", Proc. of the 29th ACL, pp.299-306 (1991).