# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 論題(和文) | 柔軟でコンパクトな純粋関数型デコーダの検討 |
| Title(English) | |
| 著者(和文) | 篠崎 隆宏, 関嶋 政和, 萩原 茂樹, 古井 貞熙 |
| Authors(English) | Takahiro Shinozaki, Masakazu Sekijima, Shigeki Hagihara, SADAOKI FURUI |
| 出典(和文) | 日本音響学会2010年秋季講演論文集, , No. 1-Q-26, pp. 181-182 |
| Citation(English) | , , No. 1-Q-26, pp. 181-182 |
| 発行日 / Pub. date | 2010, 9 |

# 柔軟でコンパクトな純粋関数型デコーダの検討*

○篠崎隆宏, 関嶋政和, 萩原茂樹, 古井貞熙 (東工大)

## 1 はじめに

大規模な統計モデルを用いた音声認識等の研究において，新しい統計モデルやその他様々な手法を試みるためには，しばしば学習や認識を行うソフトウエアの拡張が必要となる．しかしながら，これらのソフトウエアは複雑で大規模なものであり，音声認識分野への新規参入者のみならず音声認識の専門家においても全体を理解することは容易とは言えず，研究遂行上のボトルネックとなっている．

この問題に対するひとつのアプローチとして，対象とする確率モデルの一般化を行いその一般化された確率モデルを扱うソフトウエアを開発することで，そのインスタンスである様々なモデルに対応することである．例えば近年応用が広がっている重みつき有限状態トランスデューサ (WFST) やベイジアンネット (BN) はこの例とみなすことが出来る．しかしながら，このアプローチではあらかじめ対象とされる事柄の範囲外のことを行おうとすると，やはり複雑なソフトウエアの改造が必要になる問題がある．

他方，ソフトウエア工学の分野では純粋関数型プログラミングの研究が進み，純粋関数型言語およびその処理系が次世代のプログラミングパラダイムとして実用化されつつある．純粋関数型言語では関数に副作用が存在せず遅延評価との相性がよい特徴があり，高度に抽象化されたプログラムを記述することが可能である．また，関数の評価順序について処理系側の自由度が大きく，将来的には並列計算などにおいても有利と考えられている．そこで純粋関数型プログラミングを大規模統計処理ソフトウエアの開発に応用すれば，統計モデルの学習や認識処理ソフトウエアを柔軟かつコンパクトに記述でき，新しい改良を試みる上で有用と考えられる．

しかしながら純粋関数型プログラミングでは，関数を関数の引数や値として扱う高階関数がその構成要素であり，値の名前は在っても変数 (や変数への代入) が存在せず，また配列の更新が計算効率上非常に高価となるなど，従来の手続き型プログラミングとは大きく異なった特徴があり，大規模な統計モデルの処理にどのように応用できるかは自明では無い．そこで本研究では実証的に純粋関数型言語を用いて大語彙連続音声認識デコーダを記述し，その可能性について検討を行う．

## 2 純粋関数型言語

現在もっとも一般的なプログラミングパラダイムは C 言語に代表される手続き型プログラミングである．手続き型言語では一般に変数の値すなわち状態を更新しながら逐次的に手続きが実行されていく．それに対して純粋関数型言語は λ 計算の概念をプログラミング言語として体現したものである．純粋関数型言語では状態の概念が存在せず，全ての関数が副作用を持たない．この結果，遅延評価との相性が良く [1]，多くの純粋関数型言語で遅延評価がデフォルトの評価戦略となっている．遅延評価を用いることで関数間の呼出関係と実際の計算が行われるタイミングを分離することが可能となり，プログラムのモジュール性を大きく向上させることが出来る．副作用を必然的に伴う入出力についてはモナドを利用する方法が提案されており，プログラムの純粋性を保ちつつ入出力を行うことが可能となっている．純粋関数型言語の例としては Haskell や Concurrent Clean が挙げられる．

## 3 開発ソフトウエア

本研究において開発を行っている純粋関数型デコーダ「Husky」は，WFST 型の認識デコーダであり，プログラミング言語 Haskell を用いて実装されている．入力として AT&T 形式の WFST と HTK 形式の隠れマルコフモデル (HMM) の状態定義ファイル，音声特徴量ファイルのリスト，探索パラメタ等の設定ファイルを受け取り，音声認識を行うことが出来る．認識処理は 1 パス型のフレーム同期ビームサーチにより行われる．入出力はモナドを用いて実現しており，各音声特徴量ファイルを処理する毎に結果を出力する．プログラム上は音声特徴量ファイル全体を一度に読み込み探索部に渡すような構成となっているが，遅延読み出しを行うため，実際には探索の進行とともに開始フレームから逐次的に読み出される．

表 1 に Husky のコード量を示す．純粋関数型プログラミングを用いた Husky は非常にコンパクトであり，データ構造の定義や入出力の為のコードを含めても 400 行に満たない．さらに探索や main 関数その他に限れば 120 行ほどである．これは従来の手続き型言語を用いた音声認識デコーダのソフトウエアが数千から数万行のオーダーであることと対照的である．

*Development of a flexible and compact pure functional decoder. by SHINOZAKI Takahiro, SEKIJIMA Masakazu, HAGIHARA Shigeki and SADAOKI Furui (Tokyo Institute of Technology)

Table 1 Number of lines in Husky's source code

| Category | Lines |
|---|---|
| Data structure definitions | 43 |
| IO functions | 95 |
| Main and search functions, etc. | 119 |
| Comments, blank lines | 125 |
| Total | 382 |

## 4 データ構造

純粋関数型言語では変数が存在しないため，構造データの操作について手続き型言語とは異なった扱いが必要となる．例えば，純粋関数型言語では配列の一部の要素を変更することは不可能である．そのため元の配列の一部の値が更新された配列が必要な場合は，新たに配列全体を生成することとなり，配列の更新が非常に高コストとなる．他方線形リストのようなデータ構造への要素の追加は，表向きは全要素をコピーしつつも，実際には舞台裏でデータ構造の大部分を更新前のデータと共有することが出来，メモリや CPU の利用効率が高い．またサイズ平衡 2 分木等の木構造を用いる場合も要素の更新に対するコストが安価であり，純粋関数型プログラムにおいて多用される．配列もまったく用いられない訳では無く，一度初期化した後に参照のみを行う場合は定数コストで高速な参照を行えるメリットがある．これらを踏まえ，Husky では WFST ネットワークや HMM 状態出力確率定義を配列，特徴量ベクトル系列を線形リスト，ビーム探索時の仮説候補を木構造を用いて表現している．

また探索時において，一般に HMM の同じ状態の同じ特徴量ベクトルに対する尤度が複数回使用されることから，計算効率上何らかのキャッシュ機構が必要となる．純粋関数型プログラミングでは処理系によっては自動的な最適化が期待できる場合もあるが，そうでない場合は配列要素の遅延評価を利用して計算結果をメモ化することや，あるいはそれと同等な機能を実現するライブラリ関数を利用することが考えられる．今回は後者の方法を用いた．

## 5 音声認識実験

使用した音声特徴量は MFCC12 次元と対数エネルギー，およびそれらのデルタ項とデルタデルタ項の計 39 次元である．音響モデルは 3000 状態状態共有混合ガウス分布トライホンモデルであり，日本語話し言葉コーパス CSJ [2] の学会講演音声より最小音素誤り基準 (MPE) により学習した．学習データ量は 254 時間であり，各状態の混合数は 32 である．言語モデルは CSJ の学会および模擬講演 6.8M 単語から学習したトライグラムモデルであり，辞書サイズは 30k である．WFST は，HMM 状態遷移，トライフォン音素コンテキスト，発音辞書，トライグラム言語モデルを AT&T ツールキットを用いて合成して作成した．テストセットは男性話者による学会講演 10 講演からなる CSJ 評価セットである．Haskell 処理系には GHC ver 6.10.4 を用いた．

Table 2 Word accuracies for the CSJ test set

| Lecture | Word accuracy |
|---|---|
| A01M0097 | 93.0 |
| A01M0110 | 92.1 |
| A01M0137 | 82.5 |
| A03M0106 | 71.4 |
| A03M0112 | 88.8 |
| A03M0156 | 69.4 |
| A04M0051 | 88.2 |
| A04M0121 | 78.0 |
| A04M0123 | 81.5 |
| A05M0011 | 75.4 |
| Average | 81.1 |

CSJ 評価セットに対して認識評価を行った結果を表 2 に示す．不特定話者認識システムにおいて平均単語正解精度 81.1%が得られており，$T^3$ デコーダと同等の良好な結果が得られた．他方，現時点での Husky の実装では，必要メモリ量が 40 Gbyte，実行時間が実時間の 80 倍程と大きい値となった．計算コストが高い理由の一つは，関数間でのデータの受渡しの背後でメモリの確保とガーベージコレクション (GC) が繰返されることである．ストリーム融合法の活用の他，グラフ構造の操作など汎用性のある関数の外部実装化やデータ構造の最適化などを行うことで，メモリと計算量は今後大幅に削減出来ると考えられる．

## 6 まとめ

純粋関数型デコーダを実装し，大語彙連続音声認識実験において評価を行った．

**謝辞** 本研究は JST 研究シーズ探索プログラム (融合分野) の助成を受けたものである．

## 参考文献

[1] G. J. Sussman et al., "Structure and Interpretation of Computer Programs," MIT press, 1984.

[2] T. Kawahara et al., SSPR2003, pp. 135–138, 2003.