様式 C-19

科学研究費補助金研究成果報告書

平成２１年４月２０日現在

研究種目：若手研究（スタートアップ）
研究期間：２００７～２００８
課題番号：１９８００６９
研究課題名（和文）　ユーザ指向の探索アルゴリズムライブラリの構築

研究課題名（英文）　Development of a User-Oriented Search Algorithm Library

研究代表者
森本　武資（MORIMOTO TAKESHI）
島根大学・総合理工学部・助教
研究者番号：８０４５５４７７

研究成果の概要：
探索問題の仕様に相当するプログラムにわずかな記述を追加するだけで，枝刈りを行う実行効率のよい探索プログラムの開発を可能にする，ユーザ指向の探索アルゴリズムライブラリを構築した．従来研究におけるライブラリ構築の基礎となっていた遅延データ構造 Improving Sequence を拡張することにより，従来は実装の難しかったアルゴリズムについても，ユーザ指向のライブラリ関数として実装できることを実証した．

交付額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	1,360,000	0	1,360,000
２００８年度	1,350,000	405,000	1,755,000
年度			
年度			
年度			
総　計	2,710,000	405,000	3,115,000

研究分野：総合領域
科研費の分科・細目：情報学・ソフトウェア
キーワード：関数型言語，　探索問題，　ユーザ指向

１．研究開始当初の背景

プログラム開発の目的は，プログラムによって問題を解くことであるから，解こうとしている問題がどのようなものか（仕様）を記述するだけで，問題を解けるのが理想である．しかし，現実には，問題を解くためにはどうすればよいのか（実装）も記述する必要がある．

実装にかかる負担を減らすための典型的な手法の一つに，実装のライブラリ化がある．ライブラリの利用者は，典型的な実装のための煩雑なプログラムをしなくてもよい．

しかし，従来のライブラリの多くには，プログラムの自然な記述を妨げてしまうという問題がある．なぜなら，ライブラリの利用者は，ライブラリを利用できるような形に，プログラムを書き下すことを強制されるからである．

本来，実装というのは，仕様という目的を満たすための，手段である．しかし，従来のライブラリにおいては，手段（実装）のため

に目的（仕様）を歪めるという，主従関係の逆転が起こる．そのような，いわば「ライブラリ指向」のライブラリは，利用に手間がかかり，プログラムの形を歪にするため，プログラムの生産性と可読性を低下させる．

２．研究の目的

本研究では，この問題を解決するため，「ユーザ指向」のライブラリ構築を目的とする．すなわち，仕様（に相当するプログラム）の簡潔さを保ったまま利用可能な，柔軟性の高いライブラリを構築する．ライブラリ指向のライブラリが，ライブラリへの配慮をプログラマ（ユーザ）に要請していたのに対し，ユーザ指向のライブラリは，そのような配慮を要請しない．

構築するライブラリの対象を，本研究では，探索アルゴリズムに絞る．その理由は，探索アルゴリズムの利用範囲の広さと利用効果の高さにある．探索アルゴリズムが，実世界で頻繁に遭遇する探索問題を実行効率よく解くための，必須技術であることは論を待たず，ライブラリ構築の意義は大きい．

３．研究の方法

ユーザ指向のライブラリ構築の鍵は，遅延リストに似た遅延データ構造 Improving Sequence を用いる点にある．

このデータ構造を用いると，探索問題の仕様に相当するプログラムにわずかな記述を追加するだけで，探索空間全体に相当する巨大な遅延データ構造を構成することができる．

遅延データ構造とは，データ構造の各要素を，実際に必要とするまでは計算しない（遅延する）データ構造である．ゆえに，遅延データ構造である Improving Sequence を用いれば，探索空間の一部だけを，必要に応じて参照しながら探索することによって，不要なデータ構造を生成することなく，解を得ることができる．

探索空間のどの部分を必要とし不要とするかという判断は，Improving Sequence 上の関数としてモジュール化できるので，探索アルゴリズムに基づいて判断を行う関数を予めライブラリとして用意できれば，アルゴリズムに基づいて枝刈りを行う実行効率のよい探索プログラムを，容易に得ることができる．

我々は，従来研究において，古典的な三つの探索アルゴリズム（最良優先，深さ優先分枝限定，反復深化）に基づく関数を実装し，ユーザ指向の探索アルゴリズムライブラリの実現可能性を示してきた．このライブラリは，広く使われている関数型言語 Haskell の標準的な処理系 GHC 上で，処理系を一切拡張することなく構築できた．本研究は，これらの従来研究を，実用的な技術となるよう発展させることを目指すものである．

その実現に向け，本研究では，以下の 3 つを具体的な研究目標として定めた．

(1) 探索アルゴリズムの実装の開拓

探索問題を解く際にはその問題に適した探索アルゴリズムを利用することが重要である．そのため，本研究では，比較的新しい（すでに実現済みの古典的なものとは趣の異なる）探索アルゴリズムである LDS（Limited Discrepancy Search）を実装する．

(2) 例題の開拓

提案ライブラリの有効性を実証するために，提案ライブラリを用いてより多くの例題を解く．例題を蓄積することには，単に提案ライブラリの有効性を確かめるという以外にも，ライブラリの利用者に対して，利用方法の具体例を提示できるという点でも意義がある．

(3) アルゴリズムの実装手法の効率化

徹底的なユーザ指向は，利用者に利便性を与える反面，開発者に負担がかかる可能性がある．なぜなら，開発者は，ライブラリ指向で考案された通常のアルゴリズム記述を，ユーザ指向の記述に書き換える必要があるからである．アルゴリズムの実装経験を蓄積することにより，この書き換えを容易にするための知見を得る．

４．研究成果

本研究の成果は以下の通りである．まず，ライブラリ構築の基礎である，遅延データ構造 Improving Sequence を拡張し，不要な計算を生ずることなく，データ構造の終端を確認できるようにした．次に，この拡張が以下の 3 つの特長：

- ライブラリの適用範囲の拡充
- ライブラリの実装に関する記述性の向上
- ライブラリの利用に関する記述性の保持

を備えることを明らかにした．

これら 3 つの特長は，前節で述べた本研究の 3 つの研究目標に対応するものである．すなわち，1 つ目の特長である，適用範囲の拡充は，研究目標(1)探索アルゴリズムの実装の拡充，および，(2)例題の開拓に対応する．また，2 つ目の特長である，実装に関する記述性の向上は，研究目標(3)アルゴリズムの実装手法の効率化に対応する．さらに，3 つ

目の特長である，利用に関する記述性の保持は，上述の 2 つの特長を享受しつつも，本研究の主目的であるユーザ指向ライブラリの構築が実現できることを意味する．

すなわち，本研究の成果とは，Improving Sequence の拡張により，従来よりも多くのアルゴリズムを備えたライブラリを，従来と同様の高い柔軟性を保ったまま構築したこと，とまとめられる．

本研究の成果の詳細に立ち入る前に，構築したライブラリに基づく探索プログラミングの様子を，具体例に沿って例説する．（ただし，ここでは，ライブラリの使用方法の全体像のみを説明し，プログラムそのものの説明は省略する．）

まず，解きたい探索問題の仕様（に相当するプログラム）を，例えば図 1 のように，記述する．

```
search state
  | isGoal state = cost state
  | otherwise    =
    minimum [search s | s <- children state]
```

図 1：探索問題の仕様の記述例

次に，この仕様にわずかな記述を追加する（図 2）．この追加だけで，枝刈りを行う実行効率のよい探索プログラムを構成できる．

```
search minium' search' state
  | isGoal state = Done (cost state)
  | otherwise    = Cont (eval state) $
    minimum' [search' s | s <- children state]
```

図 2：提案ライブラリの使用例

図 2 の青色の部分が，ライブラリを使用するために追加した記述である．ユーザ指向ライブラリの高い柔軟性により，元のプログラム（図 1）の構造を保ったまま，ライブラリを使用できていることに注意されたい．

なお，追加内容の大部分は，半ば機械的に記述することができる．実際，minimum’と search’は，元々の minimum と search をパラメタ化しただけである．また，Done は探索が終了する場所に挿入しただけであり，Cont は次の（再帰的な）探索の手前に挿入しただけである．利用者の工夫が必要なのは eval state の記述のみであり，個々の探索問題に適したヒューリスティック関数 eval を考案する必要がある．

こうして得られたプログラム search に対して，例えば toBF search とすれば最良優先（Best First）に基づく探索プログラムが得られ，toID search とすれば反復深化（Iterative Deepening）に基づく探索プログラムが得られる．

本研究では，新しく，LDS アルゴリズムに関するライブラリ関数 toLDS を実装した．toLDS を，toBF や toID などと共に，一つの枠組みの中で実装できるようにしたことが，本研究の成果である．

さて，LDS の実装に先だって，本研究では，まず，Improving Sequence を拡張する．本研究で拡張した Improving Sequence は，関数型言語 Haskell においては，図 3 のように定義される．

```
data Ord a =>
     IS a = Done a
          | Cont a (IS a)
```

図 3：遅延データ構造 IS の定義

このデータ構造 IS では，Improving Sequence の終端をデータ構成子 Done によって表現する．ゆえに，パターンマッチのみで，終端か否かを判定できる．

一方，従来の Improving Sequence には，Done に相当するデータ構成子はなく，終端の値も，中途の値も，まとめて，データ構成子 Cont によって表現する．ゆえに，後続の要素（IS a）を参照してみないことには，Improving Sequence が終端に達したかどうかを，判定できなかった．

「３．研究の方法」で述べたように，遅延データ構造における（後続の）要素は，参照されるまでは生成されないので，どこからも参照されない要素は，そのまま枝刈りされるはずのものである．しかし，終端判定のためだけに，後続の要素を参照してしまうと，本来は枝刈りされるはずの不要な要素も生成することになり，実行効率が低下してしまうという問題点が，従来の Improving Sequence にはあった．これに対して，考案した拡張版は，後続の要素を参照せずとも終端を判定できるので，実行効率がよい．

Improving Sequence の終端判定を，簡潔かつ実行効率よく実現できることは重要である．なぜなら，Improving Sequence に基づく探索アルゴリズムの実装という文脈においては，Improving Sequence の終端判定は，探索空間の終端判定（すなわち，探索解の一つに到達したか否かの判定）に相当するからである．様々なアルゴリズムにおいて，この判定は必須のものであるから，ライブラリ構築のためのプリミティブとして用意するだけの価値があることが，本研究を進める過程で明らかになった．さらに，考案した図 3 の拡張は，先に述べたように，3 つの特長を併せもつことも明らかにすることができた．以下，それぞれの特長について説明する．

(1) ライブラリの適用範囲の拡充

拡張した Improving Sequence の１つ目の特長は，実装可能なアルゴリズムの幅を広げられることである．実際，LDS アルゴリズムは，従来の Improving Sequence では，実行効率のよい実装を記述できなかったが，この拡張により（実行効率のよい）実装が可能になった．これは，上で述べたように，終端判定が，LDS の実装にとって本質的なものだからである．

さらに，我々は，実装した LDS を用いることにより，新たな例題として Job Shop Scheduling 問題を，仕様を記述するだけで，解くことができた．

(2)ライブラリの実装に関する記述性の向上

拡張した Improving Sequence の２つ目の特長は，ライブラリの実装を，従来よりも，簡潔に記述できることである．例えば，反復深化アルゴリズムは，従来の Imporoving Sequence でも実装することはできたが，拡張版を用いた方が実装が簡潔になった．理由は，これも，終端判定の記述にある．反復深化アルゴリズムの実装においては，LDS とは異なり，従来の Improving Sequence でも終端判定を（不自然で技巧的ではあるが）実行効率よく記述できていた．拡張版では，この技巧的な部分を，自然な記述に置き換えることで，実装を簡潔にすることができた．

(3)ライブラリの利用に関する記述性の保持

これまでに述べた２つの特長を享受しつつも，図２で示したように，従来の Improving Sequence に基づくライブラリと同様の，柔軟性の高いユーザ指向ライブラリを構築することができた．

以上のように，本研究では，徹底したユーザ指向と，様々な探索アルゴリズムの蓄積により，探索プログラムの生産性と可読性を大幅に高めることができた．構築したライブラリ自体が有用であることはもとより，将来的には，関数型言語以外の様々なプログラミング言語への応用を視野に入れている．本研究は，その布石であり，提案手法の課題を洗い出すことができた．

実際，本研究では，従来の Improving Sequence の終端判定に関する問題点とその解決方法を明らかにすることができた．そして，考案した拡張版 Improving Sequence が，従来版の優れた性質をそのまま引き継いだ，自然な拡張であることを確かめることができた．

本研究を学術的な観点から見れば，記述性と実行性能を同時に実現するという，プログラミング言語研究の主要課題の一つについて，有望な方向性を示唆しているといえる．また，関数型言語研究の側面から見ると，本研究は，世界的にもほとんど例がない用法によって遅延データ構造を効果的に活用しており，関数プログラミングの可能性を押し広げるものである．

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計 １ 件）

① 小池勇治，西岡真吾，森本武資，丸川雄三，高野明彦 ： 分散連想計算サーバー群を統合する連想検索システム「想・IMAGINE」，情報処理学会 自然言語処理研究会，２００８年７月１７日，公立はこだて未来大学．
http://ci.nii.ac.jp/naid/10021838558

６．研究組織

(1)研究代表者
森本　武資（MORIMOTO TAKESHI）
島根大学・総合理工学部・助教
研究者番号：８０４５５４７７