

n 天王問題を解く決定性と非決定性のプログラムに関する考察

組込みデザイン研究室 佐野勇樹 (指導教員 島和之)

1 はじめに

ソフトウェアの生産性と品質の向上のため、再利用可能なモジュールを組み合わせることでソフトウェアを開発することが重要である。関数型プログラミング言語では、高階関数や遅延評価によって関数をモジュールとして組み合わせることができる。このため、リストや木構造の操作、数値計算、 α β 法などのアルゴリズムを再利用可能なモジュールとして記述し易い [1]。

ある状態と入力の組に対して複数の状態遷移が定められているオートマトンを非決定性オートマトンという [2]。非決定性量子コンピュータが研究されているが、まだ実用化されていない。決定性のコンピュータは、バックトラックや並行処理によって非決定性をシミュレートできる。そこで、非決定性のシミュレーションを再利用可能なモジュールとして実装しているプログラムを非決定性プログラムと呼ぶ。

本研究では、非決定性プログラムの有用性を確かめるため、 n 天王問題を例題とし、決定性と非決定性の解法プログラムの構造、時間計算量、空間計算量について考察する [3]。 n 天王問題とは、一般化四天王問題 [4] の人数を n とし、強さに関する与えられた条件から強い順に並べる問題である。この問題は制約充足問題の一種であり、スケジューリング問題への応用が考えられる。

2 非決定性プログラム

Scheme 言語と Hat 言語は、言語仕様として非決定性計算のための文法や特別な機能は持たないが、継続を用いたバックトラックによって非決定性計算をシミュレートできる。Hat 言語では全ての関数呼び出しが継続渡しスタイルであり、呼び出された関数で継続を利用できる。継続渡しスタイル (Continuation Passing Style, CPS) では関数からの復帰において計算結果を継続に渡す [5, 6]。

リスト 1 は Hat 言語で非決定性計算をシミュレートするための関数の定義である [4]。関数 `forEach` は、通常の引数 `list`、`action` と継続の引数 `return` を受け取る。`list` の各要素を一つずつ選び、その要素を第 1 引数、それ以外の要素からなるリストを第 2 引数とし、`action` を呼び出すことを繰り返す。全ての要素を選び終わったら、`return` で示される継続へと戻る。

関数 `amb` は、通常の引数 `list` と継続の引数 `cont` を受け取る。`list` を `forEach` の `list`、`cont` を `forEach` の `action` へと渡す。すなわち、`forEach` は `list` の各要素を一つずつ選び、選んだ要素と残りのリストを `amb` の `cont` へ繰り返し渡す。このため、`amb` の呼び出し元では、`amb` が `list` の要素を非決定的に一つ選び、その要素を第 1 戻り値、それ以外の要素からなるリストを第 2 戻り値として返すように見える。

リスト 1 Hat 言語における非決定性計算のための関数定義

```
1 (defineCPS forEach ^ (list action . return) fix
2   (^ (loop back rest) 00o1l1|2Zz5Ss9gq--_-'")
3   unless (isPair rest) return ^()
4   splitPair rest ^ (el rest)
5   moveAll back rest ^ (others)
6   action el others ^() makePair el back ^ (back)
7   loop back rest ) ( ) list)
8 (defineCPS amb ^ (list . cont) forEach list cont)
```

表 1 計算量の比較

| 言語 | C | Scheme | Hat |
|-------|---------------------|----------------------|----------------------|
| 時間計算量 | $\mathcal{O}(mn^n)$ | $\mathcal{O}(mn^n)$ | $\mathcal{O}(mn!)$ |
| 空間計算量 | $\mathcal{O}(m+n)$ | $\mathcal{O}(m+n^2)$ | $\mathcal{O}(m+n^2)$ |

3 n 天王問題

n 天王問題では、 n 人 A, B, C, ... の強さの順位 (強いほど小さい) を、それぞれ a, b, ... とおき、以下の書式で条件を入力し、条件を満たす順列を全て出力するものとする。

(左辺) (演算子) (右辺) (コメント)

(左辺) と (右辺) は a, b, ... または 1, 2, ..., n のいずれかである。(演算子) は $<$, $>$, $=$, $<=$, $>=$, $!=$ のいずれかである。(コメント) は無視される。(左辺), (演算子), (右辺), (コメント) の間には 1 つ以上の空白が必要である。

4 考察

n 天王問題の解法プログラムに m 個の条件を与えたときの時間計算量 (解を見つけるための計算時間) および空間計算量 (解を見つけるために必要なメモリ量) を表 1 に示す。

C 言語の解法プログラムでは、順位割り当てのため for 文を n 個重ね、全てのパターンを生成し、各パターンに対し、最大 m 個の条件を判定する必要があるため、時間計算量は $\mathcal{O}(mn^n)$ となる。また、 m 個の条件および n 個の順位を記憶する必要があるため、空間計算量は $\mathcal{O}(m+n)$ となる。

Scheme 言語の解法プログラムでは、順位割り当てのツリーの分岐は n 、深さは n なので時間計算量は $\mathcal{O}(mn^n)$ である。また、 m 個の条件、および、分岐 n 深さ n のツリーをメモリに記憶しておく必要があるため、空間計算量は $\mathcal{O}(m+n^2)$ となる。

Hat 言語の解法プログラムでは、根ノードからの深さ d における分岐の数が $n-d$ となるため、時間計算量は $\mathcal{O}(mn!)$ である。順位割り当ての際に n 個のノードを記録し、その中から 1 つの順位を選ぶ。このため、残りのノードも記憶しておく必要があるため、空間計算量は $\mathcal{O}(m+n^2)$ となる。

5 まとめと今後の課題

本研究では、C 言語、Scheme 言語、Hat 言語による n 天王問題の解法プログラムについて考察した。今後の課題としては、Prolog など他言語との比較やブロック構造の簡略化などが挙げられる。

参考文献

- [1] J. Hughes: Why functional programming matters, The Computer Journal - Special issue on Lazy functional programming archive, vol.32, no.2, pp.98–107, April 1989.
- [2] 藤原曉宏: はじめて学ぶオートマトンと言語理論, 森北出版株式会社, 2015 年 7 月.
- [3] 佐野勇樹, 横本勇希, 小島渚, 島和之: 継続渡しスタイルの関数型言語における非決定性プログラムに関する考察, 電子情報通信学会技術研究報告 (SS2020-13), vol.120, no.193, pp.49–54, 2020 年 10 月.
- [4] 横本勇希, 小島渚, 島和之: 四天王問題の一般化と継続渡しスタイルの関数型言語における非決定性計算による解法プログラム, 電子情報通信学会技術研究報告 (SS2019-18), vol.119, no.246, pp.31–36, 2019 年 10 月.
- [5] G.J. Sussman and G.L.S. Jr.: Scheme: A interpreter for extended lambda calculus, Higher-Order and Symbolic Computation, pp.405–439, Nov. 1998.
- [6] J.C. Reynolds: The discoveries of continuations, Lisp and Symbolic Computation, vol.6, pp.233–247, Nov. 1993.