正規表現のあいまいさ除去戦略の効率的実現*
（プロジェクト 52b 年次報告）

奥居　哲
中部大学工学部
情報工学科

鈴木　大郎
会津大学コンピュータ理工学部
（学外協力者）

1　はじめに

正規表現を用いた照合技術は，テキスト・データの処理に欠かせない情報処理技術である．近年では，ネットワークの分野における侵入検知のようなバイナリ・データの処理や，バイオ・インフォマティクスの分野においても，その有用性が高まっている．

正規表現を用いた照合で重要なのが，照合によって生じる別解をどのように解消するかという，あいまいさ除去（disambiguation）の問題である．POSIX 1003.2 標準規格 [1] では，照合のあいまいさを，いわゆる最左・最長規則（leftmost-longest rule）にしたがって解消するように求めているが，実際には，これに厳密に準拠している処理系は，ほとんどないのが現状であり，大抵は，貪欲（greedy）照合と呼ばれる方法であいまいさを解消している．これは，（1）POSIX の最左・最長規則の定義自体が複雑で，形式化されていないこと，（2）現在主流のバックトラックに基づく照合によっては，効率的実現が困難であることが要因である．

また，現在主流のバックトラックに基づく照合手法には，場合によっては計算爆発を引き起こすという問題もある．このことはユーザが処理に要する時間をあらかじめ見積もることを困難にする他，DOS 攻撃等のサイバー攻撃に弱いという欠点を持つ．

そこで，最左・最左規則にしたがう照合をバックトラックに拠らずに効率的に実現する新たな手法を明らかにしようというのが，本研究プロジェクトの目的である．

* 本研究の遂行にあたっては科研費（基盤 (C) 22500019）の助成を受けている．

2　進捗

本研究は，（1）基本的なアルゴリズムを明らかにし，その正当性を証明する段階と，（2）最適化及びコンパイル手法を明らかにする段階に分けて 3 年計画で進められている．現在，その第 1 年度であり，（1）の段階がほぼ終了したところである．特に以下の点については既に終了し，公表済みである [7].

1. 最左・最長規則に基づくあいまいさ除去を構文解析木を用いて簡潔かつ形式的に表現すること
2. 最左・最長解を与える非決定性オートマトン（NFA）を Glushkov オートマトン [4]（ポジジョン・オートマトン [6]）の拡張に基づき構築する手法を明らかにすること
3. 部分集合構成に基づき，上記の NFA 上でバックトラックに拠らない解の探索を行うアルゴリズムを構築すること

現在，このアルゴリズムの正当性の証明がほぼ終わり，投稿準備中である．

3　今後の計画

第 2，第 3 年度の研究はそれぞれ以下のように進めていく計画である．

3.1　第 2 年度

基本アルゴリズムの静的な最適化とそれに基づくコンパイル手法を明らかにする．実用に用いられる正規表現の中には，あいまいさの少ないものも多い．よっ

て，与えられた正規表現の特殊性を考慮して，NFA をあらかじめ，仮想機械のコードに翻訳することで，実行効率を高める．このために必要となる仮想機械の特性について明らかにする．

3.2 第３年度

基本アルゴリズムの動的な（実行時の）最適化手法を明らかにする．正規表現は，反復のパターンを多く含むため，実行時に得られるプロファイル情報を基にして，より効率的に実行可能なコードを実行時に生成しようというものである．

4 関連研究の動向

本研究の発表と前後して，関連する研究結果が相次いで発表されている．中でも，Cox による Google の新しい正規表現エンジン RE2[2] は，バックトラックに拠らない照合アルゴリズムを提案しているという点で，本研究と共通している．RE2 と本研究が大きく異なるのは，あいまいさ除去の戦略である．本研究が POSIX 標準の最左・最長戦略に準拠しているのに対して，RE2 は，（現在主流のバックトラックに基づく正規表現エンジンと同じく）貪欲戦略を用いている．また，手法的な違いとしては，本研究が Glushkov オートマトンの独自拡張に基づいているのに対して，RE2 は Thompson オートマトンに基づいている．本研究が Glushkov NFA を用いているのは，ε-遷移の循環に起因する問題 [3] が根本的に解決されるからである．

一方，Le Maout[5] は，バックトラックを用いる代表的な正規表現エンジンと RE2 の性能比較を行っており，バックトラックに拠らないアプローチの有効性を明らかにしている．このことから，今後，（後方参照を必要としない場合においては）非バックトラックの実装が一般的になっていくものと思われる．

また，実装の完成度では RE2 は突出しており，本研究の現在の試験実装よりも，概して実行速度で優っている（もっとも，貪欲戦略は POSIX 最左・最長戦略より遥かに実現が容易であるので，公平な比較ではないが）．よって，本研究の今後の焦点は，EE2 に比類する実行速度を得ることの可能な最適化コンパイル手法を開発していくことである．

参考文献

[1] The Open Group Base Specification Issue 6 IEEE Std 1003.1 2004 Edition. `http://www.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap09.html`, 2004.

[2] R. Cox. Regular Expression Matching in the Wild. `http://swtch.com/~rsc/regexp/regexp3.html`, 2010.

[3] A. Frisch and L. Cardelli. Greedy Regular Expression Matching. In *ICALP04 (LNCS 3142)*, pages 618–629, 2004.

[4] V. M. Glushkov. The Abstract Theory of Automata. *Russian Mathematical Surveys*, 16(5):1–53, 1961.

[5] V. Le Maout. Regular Expressions at their Best: A Case for Rational Design. In *CIAA2010 (LNCS 6482), To apear*, 2011.

[6] R. McNaughton and H Yamada. Regular Expressions and State Graphs for Automata. *IEEE Transactions on Electronic Computers*, 9:39–47, 1960.

[7] S. Okui and T. Suzuki. Disambiguation in Regular Expression Matching via Position Automata with Augumented Transitions. In *CIAA2010 (LNCS 6482), To apear*, 2011.