パイプライン構造を有するNFAハイブリッドアーキテクチャにおける正規表現パターンの実装と評価

片庭悠介† 山口喜教‡ 前田敦司‡ 飯星貴裕‡
筑波大学第三学群情報学類† システム情報工学研究科‡

1. はじめに

本研究では、ネットワーク上を流れるストリームデータを正規表現を含んだパターンで検索することを目的に、FPGA を利用した再構成可能なハードウェアアーキテクチャを構成することを目的にする。そのために、まず Moscola[1]らの提案によるパイプライン構造と NFA ステートマシンを組み合わせたハイブリッドアーキテクチャを用いたハードウェア実装と正当性の評価を行う。本研究の目標は回路規模の縮小のために辞書圧縮の概念を用いて同じ部分文字列に対する回路を共有することで回路の効率化を図っていく[2]ことを正規表現に対応させることであるが、本稿ではその前段階として、一般的な正規表現の回路化について述べる。

本稿では拡張 NFA を定義する。従来の NFA に、index というパイプラインのタイミング情報を付属させることによって、回路の遅延を表現できるように拡張を行う。またその拡張 NFA を元にした実際の回路の構成方法を提案する。

2. ハイブリッドアーキテクチャ

Moscola が提唱したハイブリッドアーキテクチャは、入力された文字がデコードされた 1 ビット情報を 1 クロックごとにパイプライン的に伝搬することによって、任意の時点での文字入力情報を取り扱うことができるという特性を持っている。そのために、閉包や和といった文字長が一定ではない正規表現を扱う際に特に有用な手段であるといえる。図 1 にハイブリッドアーキテクチャで用いられるパイプラインと"aaabbccc"のチェーン構造のマッチング回路を示す。

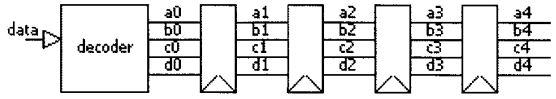

図 1(a).デコード済みパイプライン

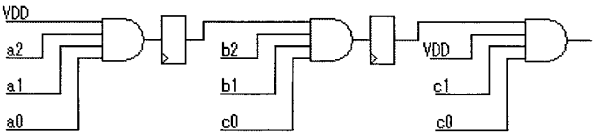

図 1(b).チェーン構造のマッチング

3. 正規表現と NFA

3.1 サポートする正規表現

本研究では POSIX の正規表現中で使われている演算子の中で、連結、閉包および和について扱うものとする。連結と閉包の拡張である"+"と"?"についてはここでは未対応であるが、演算子の等価変換を行うことで対応が可能になる。また、ドット演算子などの文字自体に関する演算子については本稿では扱わないものとする。

3.2 正規表現から提案する拡張 NFA への変換

NFA からハイブリッドアーキテクチャに適した回路を生成するための前段階として、Hopcroft[3]の正規表現の NFA 変換の拡張を行う。実際の回路ではゲートで処理できる文字数限界とクロックによる遅延の概念が存在するため、Hopcroft の NFA に遅延を考慮するための新たな index という概念を導入する。

index は回路を構成するための情報を表すものであり、文字の index はパイプラインからの取得タイミング、ε の index はパイプラインの遅延のレジスタの設置個数を意味する。図中では同一 index として扱うが厳密には意味が異なるため、アルゴリズムではこれらを区別できなくてはならない。文字の index はタイミングの移動によって変動するが、本研究では and ゲート毎に 0 になるよう決定するものとする。

また正規表現の連接を変換した、NFA 表現において、ある状態からの遷移が一方向に数珠つなぎになっている場合、それを and ゲートの入力数(n)に応じて文字をまとめることとする。こ

The implementation mechanism of the scalable hybrid architecture for regular expression patterns and its evaluation.

Yusuke KATANIWA†, †College of Information Sciences, Third Cluster of Colleges, University of Tsukuba
Yoshinori YAMAGUCHI‡, Atsushi MAEDA‡ and Takahiro IIHOSHI‡, ‡Graduate School of System and Information Sciences, University of Tsukuba

Copyright © 2010 Information Processing Society of Japan.
All Rights Reserved.

の時 NFA の発火状態のための制御ラインが必要となるので、まとめることができる連接の遷移は and ゲートの入力数 n から 1 を引いたものとなる。この時入力数 n に制限はないが、n=2 のときは文字列化することができないので、通常の NFA に index が付属しただけのものとなる。図 2,3 はこのような変換の例を示す。図 2 は正規表現"z=(aaa|abc)*d"に対応した NFA であり、図 3 はそれから得られる拡張 NFA である。

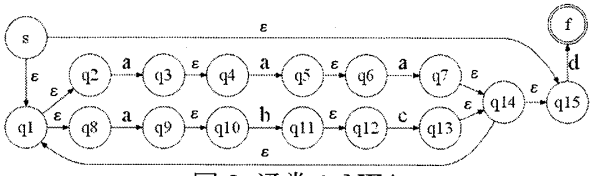

図 2. 通常の NFA

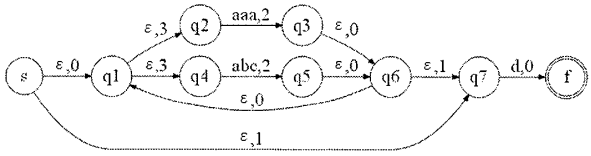

図 3. 4 入力 index 情報付属の拡張 NFA

図において各遷移には(遷移文字列,index)のペアが付加情報としてつけられる。ここにおいて index は遷移文字列が ε 以外につけられる場合は文字列の文字数から 1 を引いたもの、空列 ε の場合は遷移先の次の遷移の文字列の文字数になる。つまり文字列"aaa"の index は 2 になり、次の遷移が"aaa"である ε の index は 3 になる。この時、和と閉包により複数の遷移が存在することがあり得る。しかし、和と閉包による遷移の分岐と合流は Hopcroft の NFA 変換に従うと ε によってしか認められていない。拡張 NFA は Hopcroft の NFA の構造から連接にのみ変更を行うものなので、複数の遷移先がある場合は自動的に ε の index は 0 となる。

4. 回路の作成

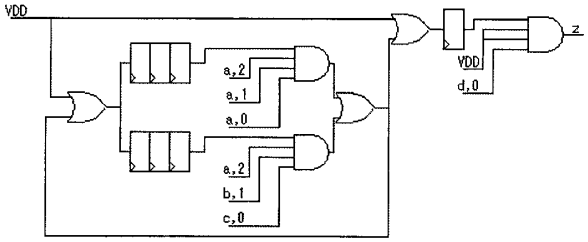

図 4.　正規表現"z=(aaa|abc)*"の回路図

拡張 NFA には、ハイブリッドアーキテクチャのチェーン構造を生成するための情報がまとめられているため、回路作成は容易であり、以下の手順に従って行うことができる。

i）状態の遷移先が複数ある場合、回路上で分岐を設置する。

ii）各 index に従って ε によるレジスタの設置と文字列によるパイプラインの接続を行う。

iii）複数の状態遷移元からの遷移がある場合、or ゲートを設置する。

iv）文字の入力数に対して and ゲートの入力数が少ない場合は VDD、or ゲートの入力数が少ない場合 GND を接続する。

前章で付属させた index 情報に基づいて図 3 の拡張 NFA を元に回路を作成したものが図 4 である。

5.正当性の評価

本研究の回路は文字列ごとに index を 0 にすることでタイミング制御を行っているが、Moscola の提唱するアーキテクチャにおいては、最終文字が 0 になるようにのみ index を調節している。レジスタ配置による回路規模の増大という点で本研究の今後の課題であるといえるが、実行する上で必要となるクロック数に変化は生じないため、クロック単位では Moscola の提唱するアーキテクチャと同様の性能を示すといえる。またどのようなメタ文字も index を 0 に調整を行うことができるということは、ゲートの入力数と演算子の出現重複に関係なくすべての出現パターンにおいて本研究は正規表現をサポートすることができるということも示している。

5.おわりに

本研究によってハイブリッドアーキテクチャに対する正規表現のハードウェア実装方法を一般的な手法で示すことができた。一方、レジスタの効率配置や対応演算子の拡張などの課題も存在するため、これらを踏まえて、辞書圧縮の概念を用いた回路自体の縮小を行う必要がある。今後はこれらの課題を追求していく予定である。

参考文献

[1]Moscola, J. et al. “A scalable hybrid regular expression pattern matcher”, in *Proc. of 14th IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'06)*, IEEE Computer Society Press, 2006, pp.337-338

[2] 飯星貴裕,山口喜教,前田敦司, 辞書圧縮の概念を用いた NIDS 向けパターンマッチングアーキテクチャ, 情報処理学会論文誌 コンピューティングシステム, vol. 2, No. 3, pp.163-172 2009

[3] J.ホップクロフト,J.ウルマン, オートマトン言語理論 計算論 I 、サイエンス社、1984.

Copyright © 2010 Information Processing Society of Japan.
All Rights Reserved.