

| | |
|--------------|---|
| Title | FPGAを用いたCKYパーキングの高速化 |
| Author(s) | 伊藤, 靖朗 |
| Citation | |
| Issue Date | 2003-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/1710 |
| Rights | |
| Description | Supervisor:中野 浩嗣, 情報科学研究科, 修士 |

FPGAを用いたCKYパージングの高速化

伊藤 靖朗 (110015)

北陸先端科学技術大学院大学 情報科学研究科

2003年2月14日

キーワード: CKY Parsing, FPGAs, Reconfigurable architectures, Reconfigurable computing.

本論文では, FPGAを用いた文脈自由文法に対するCKY (Cocke-Kasami-Younger) パージングを高速に実行するインスタンスに特化したハードウェアの実装法を提案する.

FPGA (Field Programmable Gate Array) とは, ユーザによって設計されたハードウェア設計を即座に埋め込むことができるプログラム可能なVLSIである. ユーザのハードウェア論理設計は, FPGAベンダの設計ツールを用いることによってFPGAに埋め込むことが可能である. 本研究では, 有用な計算を高速化するためにFPGAを使用する. とりわけ, 既存のソフトウェアアプローチより高速で効果的なFPGAベースの手法を開発することが目標である.

パージングは, パターン認識やプログラミング言語及び自然言語処理などの様々な分野のアプリケーションで用いられる. パージングの高速化は, 多くのアプリケーションにとって有益なことである. そのアプリケーションの例として音声認識が挙げられる.

本研究では, 部分計算の概念に基づき, FPGAを用いた計算の高速化を行う. 与えられた問題を解くために評価する関数を $f(y, x)$ とする. ただし, その関数は y を固定して繰り返し評価されることが多いとする. その場合, $f_y(x) = f(y, x)$ のようなインスタンスに特化した関数 f_y を評価することによって, $f(y, x)$ の計算の単純化が可能である. 我々のアイデアは, 固定した y と変数 x に対して $f_y(x)$ を計算するために最適化したハードウェアを作成することである. つまり, 次の2つの性質を満たす $f(y, x)$ が必要となる問題に対し, 問題のインスタンスに特化した手法をFPGAを用いて示していく.

1. 固定した y の値が問題のインスタンスに依存する
2. 問題を解くために様々な値をとる変数 x に対して $f(y, x)$ の値が繰り返し評価される

本論文では, 先に示したFPGAベースのアプローチを用いて, 文脈自由文法のパージングを文法を固定することにより高速化するハードウェアを示す. G を文脈自由文法, x を文字列, $f(G, x)$ をブール変数を返す関数とする. ただし $f(G, x)$ は, G が x を導出しかつ

そのときに限り TRUE を返すものとする. 文脈自由文法のパーズングを行うインスタンスに特化した手法のために, 任意の文脈自由文法 G に対する CKY パーズングを行う Verilog HDL ソースを生成するハードウェアジェネレータを作成した. このハードウェアジェネレータが生成した Verilog HDL ソースは, Xilinx 社の論理合成ツールを用いてコンパイルを行う. そして得られたオブジェクトファイルを Xilinx Virtex-II シリーズの FPGA にダウンロードすると, FPGA は文法 G を固定した $f_G(x)$ を計算する回路, つまり入力文字列 x に対して G が x を導出するかどうかを調べる回路になる. 本研究の CKY パーズングのシステムは, ホスト PC から文字列 x_1, x_2, x_3, \dots を入力し, FPGA はこれらの文字列が G で導出可能かどうか, すなわち $f_G(x_1), f_G(x_2), f_G(x_3), \dots$ の結果を返すことによって CKY パーズングを行う.

CKY パーズングは, 入力文字列 x の長さを n としたとき $O(n^3)$ 時間で計算することはよく知られている. N の部分集合 U, V 間のバイナリ演算 \otimes_G を $U \otimes_G V = \{A \mid (A \rightarrow BC) \in P, B \in U, \text{ and } C \in V\}$ と定義すると, CKY パーズングアルゴリズムは \otimes_G の評価を $O(n^3)$ 回繰り返す. よって, CKY パーズングにおいて \otimes_G の計算が実行時間の大部分を占めていることがわかる. そこで本研究では, \otimes_G の計算をハードウェア化することによって, \otimes_G の計算時間を生成規則数に依存しない $O(\log b)$ 時間にすることに成功した. ただし b は文脈自由文法の非終端記号数とする.

性能を評価するために, CKY パーズングを行うハードウェアとソフトウェアの比較を行った. ソフトウェアアプローチでは, p 個の生成規則全てを $O(p)$ 時間で調べることによって \otimes_G の計算を行う. よってソフトウェアアプローチの CKY パーズングは $O(n^3 p)$ 時間で計算可能である. 一方, FPGA を用いた手法は \otimes_G の評価を $O(\log b)$ 時間で行う. よって CKY パーズングは $O(n^3 \log b)$ 時間で計算可能である. 常に $b \leq p$ が成立するから, 理論的には本研究の提案手法はソフトウェアアプローチよりも高速である. そこで実際に Virtex-II シリーズの FPGA を用いて性能評価をし, ソフトウェアとの比較を行った. その結果, ソフトウェアに対して本研究の手法は最大で約 3,000 倍の高速化に成功した.