

## 平成 24 年度 情報工学コース卒業研究報告要旨

酒井 研究室	氏 名	加 藤 起 騎
卒業研究題目	難解言語 Malbolge のワード長の拡大とそのプログラミング支援ツール	
<p>難解プログラミング言語とは意図的にその言語でのプログラミングが困難になるように設計された言語である。このような言語のプログラミング手法を確立することは、知的財産権の保護等に役立つと考えられている。Malbolge は、数ある難解言語の中でも特に難解な言語として知られており、プログラムの作成すら困難である。しかし近年、飯澤らによって Malbolge のための中間言語として低級アセンブリ言語が設計され、そのプログラムから Malbolge プログラムを生成する低級アセンブラが構築された。また、長坂らによって Malbolge デバッガが開発され、Malbolge プログラミング環境の整備が進んでいる。しかし、低級アセンブリ言語を用いてプログラミングを行う際、メモリ不足という事態が度々発生していた。これは、低級アセンブラは低級アセンブリプログラムの 1 行分に相当する値を構築するために Malbolge の数十～数百ワードのメモリを使用するが、Malbolge は全体で <math>59049 (= 3^{10})</math> ワードのメモリしか持たないためである（1 ワードは三進数十桁:10trits）。具体的には、数値をインクリメントする低級アセンブリプログラムから生成された Malbolge プログラムは、全体の 10 分の 1 程度のメモリを消費していた。</p> <p>本論文では、この問題の解決のために Malbolge のワード長を 10trits から 20trits に拡大した Malbolge20 を提案し、<math>3^{20}</math> ワードのメモリを持つ Malbolge を構築する。また、Malbolge を対象としている低級アセンブラ、Malbolge デバッガを Malbolge20 に対応させる。</p> <p>Malbolge のワード長の拡大のために、まずは Malbolge20 が満たす仕様を定め、その仕様を満たすようにインタプリタの実装を行う。実装の上での元のインタプリタとの変更点は (1) メモリのデータ構造 (2) メモリの初期化方法 の 2 点である。(1) に関して、Malbolge ではメモリのデータ構造は 1 次元配列となっており、プログラムのロード終了時に余ったメモリ全てに対して初期化処理を行う。この処理を <math>3^{20}</math> ワードのメモリに対して行うと、一度に膨大な量のハードウェアの資源を消費するため、一般的なマシンでは実行することが出来ない。このため Malbolge20 では、メモリを 2 次元配列によりブロック化して管理し、必要になったブロックから初期化する遅延初期化を行う。(2) は、この遅延初期化を行う方法である。Malbolge の初期化処理は、フィボナッチ風に直前の 2 ワードを元に次の 1 ワードを順次計算していくことで行われる。従ってメモリの初期値は、インタプリタへの入力となるソースプログラムの最後 2 ワードによって決まる。しかし、ソースプログラムは Malbolge の性質により実行中に動的に書き換えられてしまうため、遅延初期化を行う場合、何らかの工夫を必要とする。これは、Malbolge 特有の初期化処理におけるメモリ値変化の周期性を利用し、実行開始時に予め各ブロックの最初 2 ワードの値のみを計算しておくことで解決する。</p> <p>低級アセンブラの 20trits 化においては、最初にこれまで行われていなかった低級アセンブラの仕様の明確化を行い、その上で必要となる変更を行う。低級アセンブラの変更点は、低級アセンブリプログラムの 1 行分に相当する値を構築する方法である。これまでは 3 進数 10 桁の値を状態とした状態遷移を考え、その上で最短経路問題を解くことにより、効率よく少ないワードで値を構築していた。しかしこの方法を 20trits の値に対して行うと実用的な時間では終わらないため、1 桁ずつ数値を設定し、一定の時間で構築する方法を本研究では用いる。Malbolge デバッガの 20trits 化は、新たに Malbolge20 のインタプリタを組み込むことで行う。また、新たに 2 次元配列のデータ構造に対応したメモリ表示の機能を追加し、実行時のメモリ値の詳細を確認しやすくする。</p>		