令和元年度 情報工学コース卒業研究報告要旨

関 研究室	氏　　名	大西晃
卒業研究題目	レジスタ付きプッシュダウンシステムに対する pre^* アルゴリズムの実装	

ソフトウェア自動検証の問題は，ソフトウェアの実装と仕様が与えられたとき，実装が仕様を満たすかどうかを判定する問題といえる．仕様を満たす集合は経験上あまり複雑な集合とならず，通常は正規集合で十分である．一方，実装のモデルとして再帰プログラムを表すプッシュダウンシステム (PDS) があるが，データ値を扱えないため近似が粗すぎるという欠点があった．筆者の属する研究室では，PDS にデータ値を記憶するレジスタ及びデータ値に関する等価性判定の機能を加えたレジスタ付きプッシュダウンシステム (PRDS) に着目し，その性質を考察する研究を行っている．筆者はこれらの知見をふまえ RPDS でモデル化されたソフトウェア実装の検証に関する以下の研究を行った．

レジスタオートマトン (RA) で認識されるデータ言語を，データ正規言語と呼ぶ．データ正規言語に関しては，通常の正規言語と同様に空判定問題が判定可能である，積集合などの言語演算について閉じていることが知られている．また，検証の問題は，入力としてソフトウェアの実装を表す RPDS $\mathcal{P}$， 実装の初期集合とバグの集合を表す RA $\mathcal{A}_I, \mathcal{A}_B$ が与えられたとき，$L(\mathcal{A}_B)$ の要素に $\mathcal{P}$ で到達することが出来る計算状況の集合を表す $pre^*_{\mathcal{P}}(L(\mathcal{A}_B))$ を用いて，$L(\mathcal{A}_I) \cap pre^*_{\mathcal{P}}(L(\mathcal{A}_B)) = \emptyset$ を判定すること，すなわち，$\mathcal{A}_I$ から $\mathcal{P}$ によって到達可能な計算状況で $L(\mathcal{A}_B)$ に属するものがないかどうかを判定する問題である．よって，$pre^*_{\mathcal{P}}(L(\mathcal{A}_B))$ が正則であり，それを認識する RA が $\mathcal{P}$ と $\mathcal{A}_I$ から構成できれば，検証問題が判定可能であるといえる．実際，$pre^*_{\mathcal{P}}(L(\mathcal{A}))$ を認識する RA が $\mathcal{P}$ と $\mathcal{A}$ から構成するアルゴリズムが知られている．

本研究では，このアルゴリズムの実装を行い，例プログラムに対する変換時間，最大メモリ使用量等から有効性を評価した．

RPDS の遷移規則は， $r = (q, \phi) \to (p, \varepsilon)/(p, j_1)/(p, j_1 j_2)$ の形式であり，順に pop, replace, push 規則と呼ぶ． 例えば，push 規則 $(q, \phi) \to (p, j_1 j_2)$ は，条件式 ϕ を満たすならば，状態 q から状態 p に遷移し, レジスタを ϕ を満たすように更新した後，スタックトップをポップし番号が j_1 と j_2 のレジスタの値を j_1 側がトップに来るようプッシュする規則である． 条件式 ϕ は遷移前後のレジスタの状態とスタックトップ値からなる同値関係を表しており，例えばレジスタ数が 2 の場合 $\phi = \{\{x_1, x_1', x_2'\}, \{x_2, y\}\}$ は更新前のレジスタ 1 と更新後のレジスタ 1, レジスタ 2 の値が等しく, 更新前のレジスタ 2 とスタックトップの値が等しく，同じ集合に含まれない要素同士の値は異なるという条件を表している．

RA の遷移規則は，$r = (q, \phi) \to p$ の形式で表され，RPDS においてスタックを入力テープとみなし，pop 規則のみが許される部分クラスとみなすことができる．用いる条件式 ϕ も RPDS と同様の形式で表され，スタックトップは次に読み込む入力データ値に相当する．

アルゴリズムの実装に際して，条件式 ϕ のデータ構造として，同値関係内の要素を過不足なく一周するような写像を与える方法を選択し計算量の減少を図った．例えば，x_1, x_1', x_2' が同値関係にある時，$f(x_1) = x_1', f(x_1') = x_2', f(x_2') = x_1$ となるような写像 f を保持する．

実験は，動作周波数 2.30GHz の Intel(R) Core(TM) i5-6200U CPU，メモリ 4.00GB の Windows 10 の環境で行った．入力例として使用レジスタ数5，遷移規則 2 の RPDS と遷移規則 2 の RA を用いた結果，push 規則群と replace 規則群の追加が 1 度ずつ行われ，19375 個の遷移規則を持つ RA が出力された．この入力に対する実行時間は平均して 200ms 前後，最大メモリ使用量は約 71MB で，出力がある程度大きい入力に対しても短時間・低コストで実行できるという結果が得られた．