

SAT ソルバを用いた有限メモリオートマトンの非空判定

1220375 外尾 元 【ソフトウェア検証・解析学研究室】

1 背景・目的

有限オートマトン (finite automata; FA) やその拡張は、計算機システムの簡潔なモデルとして広く用いられている。その応用例の一つとして、計算機システムの取り得る状態を網羅的に検査することで設計に誤りや不具合がないか調べるモデル検査がある。

FA は、系列の構成記号 (アルファベットの要素) が有限通りであることを前提としており、自然数などのデータ値を扱うシステムを直接モデル化することができない。FA の各状態にデータ値を扱えるレジスタを加えた計算モデルである有限メモリオートマトン (finite-memory automata; FMA)[1] は、自然数などのデータ値からなる系列を扱える一方、非空判定が決定可能であるなどの FA の良い性質を受け継いでいる。

本研究では、FMA に基づくモデル検査ツールの開発の第一段階として、SAT ソルバを用いた FMA の非空判定の実装を行う。FMA の非空判定は NP 完全であることが知られており [2]、従って、充足可能性 (SAT) 問題に帰着できる。手順としてまず、FMA の非空判定から SAT 問題への帰着方法 (SAT 符号化) を提案する。そして、FMA の例に適用し、SAT ソルバ MiniSat を用いて非空判定を行って、実用的な計算時間で判定が可能かどうか調べる。

2 FMA の非空判定

FA やその拡張において受理する系列が存在するかどうか判定することを非空 (non-emptiness) 判定と言う。

文献 [2] において以下のことが示されている：

命題 1 レジスタ数が k 、状態数が n である FMA A において、 A が少なくとも一つの系列を受理するならば、 A は長さ kn 以下の系列を少なくとも一つ受理する。

命題 2 Σ を、 A のレジスタの初期値をすべて含み、かつ要素数が k 以上の任意の有限集合とする。レジスタ数が k である FMA A がある系列 w を受理するならば、 A は w と同じ長さで Σ の要素のみからなる系列を少なくとも一つ受理する。

以上の命題から、FMA A の非空判定は、 A が有限アルファベット Σ 上の長さ kn 以下の系列を受理するかどうか調べる問題に帰着できる。

FMA A の非空判定問題を SAT 問題に帰着するには、 Σ 上の長さ kn の系列を表す命題変数群 $\{w_{l,\sigma}\}$ を用いて A がその系列 (の接頭語) を受理するという制約を論理式で表現すればよい。使用する主な命題変数は以下である。

- $w_{l,\sigma}$ 「時刻 l における入力記号が σ 」

- $s_{l,q}$ 「時刻 l における状態が q 」

- $a_{l,m,\sigma}$ 「時刻 l におけるレジスタ m の値が σ 」

これらの命題変数を用いて、以下の制約を論理式として記述する。

- 各時刻 l において、 $w_{l,\sigma} = 1$ となる σ 、 $s_{l,q} = 1$ となる q 、各レジスタ m について $a_{l,m,\sigma} = 1$ となる σ はいずれも一意に定まる
- 各時刻 l において、 A の遷移規則のいずれかを使って、時刻 $l-1$ の状態 q から $w_{l,\sigma} = 1$ である入力記号 σ によって時刻 l の状態 q' に遷移できる。かつ、同じ遷移規則によって、各レジスタ m の時刻 l における値が時刻 $l-1$ における値から決まる
- 時刻 0 の状態は初期状態である
- いずれかの時刻 l における状態が受理状態である

3 実験

図 1 の FMA に対して提案手法を適用する。受理状態に到達するためには一度初期状態まで戻る必要がある FMA である。使用した MiniSat のバージョンは 2.2.1、計算機の OS は macOS Ventura 13.2、CPU は Quad-Core Intel Core i5 2.3GHz、メモリは 8GB である。

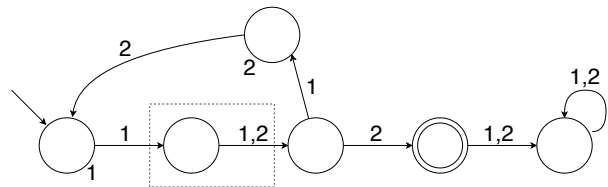


図 1 実験で用いる FMA (枠内の状態数を変更する)

レジスタ数 $k = 2$ としたとき、状態数 $n = 5, 10, 15, 20, 30, 40, 50$ に対して SAT ソルバが求解するのに要した時間はそれぞれ 0.064144 秒, 0.082928 秒, 0.105593 秒, 0.132656 秒, 0.225641 秒, 0.650651 秒であった。

4 まとめ

本研究では、FMA の非空判定を SAT ソルバで行うための SAT 符号化を提案し、FMA の例に適用した。さらに様々な FMA の例に適用し、本手法の実用性を検討することが今後の課題である。

参考文献

- [1] M.Kaminski, N.Francez, Finite-memory automata, Theoret. Comput. Sci. 134(2), 329–363 (1994).
- [2] H.Sakamoto, D.Ikeda. Intractability of decision problems for finite-memory automata. Theoret. Comput. Sci. 231(2), 297–308 (2000).