

グラフ書換え系モデル検査における 差分適用グラフ同型性判定手法の導入

坂爪 裕也 †

上田 和紀 ‡

† 早稲田大学大学院基幹理工学研究科

‡ 早稲田大学理工学術院

1 はじめに

モデル検査の状態空間爆発に対し, LMNtal [1] などのグラフ書換え系モデリング言語を用いることで, グラフ構造のもつ対称性によって状態空間の増加を緩和させることができる. しかし, 『新規状態候補と等価な状態が状態空間内に存在するか』を判定する処理(本論文ではこの処理を今後グラフ同型性判定とよぶ)の時間計算量が, グラフ構造のサイズに比例して大きくなってしまふことがボトルネックになると考えられている.

そこで, グラフ書換え系のグラフの書換え時に発生する差分(書換え差分)は書換えを行うグラフ全体のサイズに依らないため, モデルサイズに比して差分の大きさは相対的に小さくなるという特性に着目する. これを利用して, 書換え前のグラフの情報と書換え差分を利用してグラフ同型性判定を行うことで, 判定処理の効率化を図る.

グラフ書換え系向けの正規化手法として, McKay のグラフ正規化手法 [2] を最適化し, 書換え差分をもとに書換え後のグラフ正規化を行えるようにした差分適用グラフ正規化手法 [3] がある. 本研究ではこれを発展させ, 差分適用グラフ同型性判定手法としてグラフ書換え系モデル検査器に導入する.

そのうち, 本論文では導入過程で生まれた, 状態空間内の全域木に含まれる遷移の更新による, 全域木内の辺と書換え時の遷移からなる閉路の閉路長の削減手法の提案を行う.

2 閉路長の削減手法

2.1 差分適用グラフ同型性判定手法

本論文で導入する差分適用同型性判定手法は, 以下の4ステップからなる.

1. 差分適用グラフ正規化手法を用いて, 書換え前のグラフの正規化結果とグラフの書換え差分から書換え前後の正規化差分を得る.
2. 書換え前のグラフの一意表現と 1. で得た書換え前後の正規化差分から, 書換え後のグラフの一意表現と書換え前後の一意表現の対称差を得る.
3. 書換え後のグラフの一意表現を用いて, 状態空間内の同一状態候補を検索する.
4. 候補が存在すれば, 書換え前後の一意表現の対称差および状態空間内の各状態の一意表現の対称差を用いて, 書換え後のグラフと同一状態候補の一意表現の対称差を計算する.

手法内で得られるグラフの一意表現は集合の形をとっているため, 例えば書換え前の一意表現と『書換え前後の対称差』との対称差をとることで, 書換え後の一意表現が得られる.

4. において, 書換え時の遷移(候補)を含む状態空間内の閉路 C を利用する. しかし, 閉路の探索は時間計算量のコストが高い. そこで, 初期状態以外の各状態が新規状態として登録された時に生じた遷移を生成遷移とし, 生成遷移からなる木を全域木と捉える. 木における閉路の探索は最小共通祖先の探索に帰着できるため, 全域木内の遷移と書換えによって生まれる(可能性のある)遷移による閉路の探索を二分探索で実行可能である.

また, 具体的な書換え後のグラフと同一状態候補の一意表現の対称差の計算方法は, 全ての『閉路 C に含まれる各遷移がもつ, 書換え前後の一意表現の対称差』の対称差をとる, というものである. すなわち, 閉路 C の長さを削減することで, 一意表現の対称差の計算量の削減が可能である.

2.2 状態空間の全域木の更新による閉路長の削減

状態空間内の全域木の親子関係を無視し, 書換え前の状態を根とした木を考える. このとき, 状態空間内の全域木の直径は, 書換え前の状態から距離が近い状態を対象とした遷移を多く含む全域木ほど小さくなりやすい.

Incorporating Difference-based Graph Isomorphism Checking into Graph-based Model Checking

†Yuya SAKAZUME ‡Kazunori UEDA

†Graduate School of Fundamental Science and Engineering, Waseda University

‡Faculty of Science and Engineering, Waseda University

また、深さ優先で状態空間を構築している場合、距離が近い状態を対象とした遷移はより直近の状態の展開時に見つかった遷移であることが多い。以上の2点の特徴を用いて、状態空間内の全域木が全ての状態へのパスが1つであることを維持しながら、より新しく出現した遷移を全域木に採用する事で、一意表現の対称差の計算時に使用される閉路の閉路長を削減する。

状態 S_i から S_j への遷移が見つかったときの全域木の更新方法を図1に示す。ただし、 $S_k.depth$ は『生成遷移によって生成された際の S_k の深さ』である。また、 $S_k.parent$ は『状態空間内の全域木における、 S_k の親となる状態』であり、 S_k の初期値は生成遷移における S_k の書換え前の状態である。

```

if  $S_i.depth > S_j.depth$  then
     $S_j.parent := S_i.parent$ 
else if  $S_i.depth < S_j.depth$  then
     $S_i.parent := S_j.parent$ 
end if
    
```

図1 全域木の更新方法

全域木を探索した際にループが発生することを防ぐため、遷移の方向に関わらず、より浅い状態からより深い状態への遷移として扱う。また、同様の理由で、 S_i と S_j が等しい深さにある場合、その遷移を全域木に採用しない。

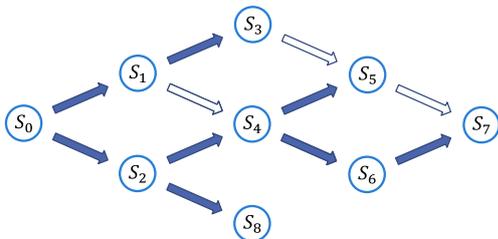


図2 全域木更新前の状態空間と全域木

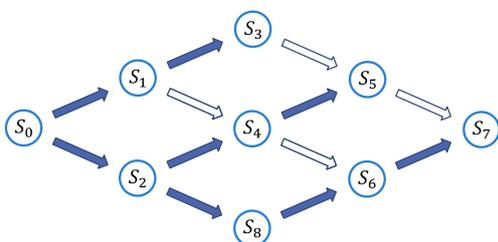


図3 全域木更新後の状態空間と全域木

一例として、図2のような状態空間と全域木があった状態で、 S_8 から S_6 への遷移が見つかった際の更新後の全域木を図3に示す。両図の矢印は遷移を示しており、中が塗りつぶされている矢印が全域木に採用されている遷移である。 S_6 の全域木上の親を S_4 から S_8 へ更新す

ることで全域木に採用される遷移を更新しているが、更新後も全域木において各状態へのパスが一意に保たれていることがわかる。

3 評価と考察

閉路長の削減効果を測るため、モデリング言語に LMNtal を使用しているグラフ書換え系モデル検査器 SLIM [4] に閉路長の探索、および図1の実装を行い、図1の有無による平均閉路長の差を測定した。例題の1つとして、ハノイの塔の問題での測定結果を図4に示す。このモデルは、モデルサイズを n とするとき、初期状態として1本の柱に n 枚の円盤が用意されている。

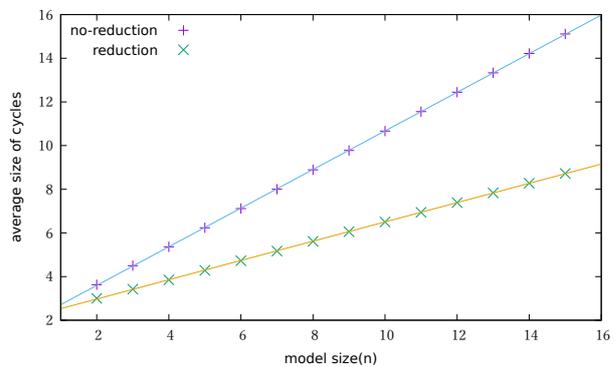


図4 ハノイの塔モデルによる実験結果

図4より、ハノイの塔モデルに関しては、モデルサイズに対する平均閉路長の増加が線形であることがわかる。また、閉路長の削減手法を導入することにより、閉路長の増加を約50%削減できているといえる。

4 まとめ

グラフ書換え系モデル検査器へ差分適用グラフ同型性判定手法を導入する過程において、『状態空間内の全域木に含まれる遷移を更新することによる、全域木内の辺と書換え時の遷移からなる閉路の閉路長の削減手法』の提案を行い、評価を行った。

参考文献

- [1] 上田和紀, 加藤紀夫. 言語モデル LMNtal, コンピュータソフトウェア, 21(2), 126-142, 2004.
- [2] McKay, Brendan D and others. Practical graph isomorphism, Department of Computer Science, Vanderbilt University Tennessee, US, 1981.
- [3] 宮原和夫, 上田和紀. グラフ書換え系のための効率的なグラフ正規化手法, コンピュータソフトウェア, Vol.33, No.1, pp.126-149, 2016.
- [4] 石川力, 堀泰祐, 村山敬, 岡部亮, 上田和紀. 軽量の LMNtal 実行時処理系 SLIM の設計と実装, 情報処理学会第70回全国大会, 2008.