4K-5

Ambient Logic モデル検査におけるプロセス式のグループ化による検証の効率化

森田　哲平†　　加藤　　暢‡　樋口　昌宏§
近畿大学理工学部情報学科

1　序論

我々は文献 [1] において，形式的なモデルに Ambient Calculus[3] を利用し，物流システムそのものをモデル化し，監視するシステムを構築した．これは，実際の物の移動と，モデル化した式の遷移を対応させながら物流におけるコンテナの移動に誤りがないか監視するものである．しかし，この式そのものに誤りが含まれていては正しい監視が行えない．そこで，文献 [2] において，物流システムで用いられているプロセス式が所期の性質を満足するか検査するシステムを構築した．しかし，単純な検査方法では状態空間爆発を起こすので，文献 [2] では，partial order reduction をほどこし，状態空間爆発を防いでいる．それでも文献 [2] のシステムでは通常の PC ではコンテナ数 150 個ほどのプロセス式の検証が限界であり，それ以上の規模を持つプロセス式の検証は不可能であった．本研究では文献 [2] で提案された対称性を利用した partial order reduction によるグループ化を改良し，さらに効率の良いモデル検査の方法を提案する．また，本研究で構築した検査システムで検証実験を行い，文献 [2] で用いられた検査システムとの性能の比較を行った．

2　Ambient Calculus を用いた物流監視システム

我々は物流監視システムのモデル化に，Ambient Calculus を用いている．Ambient Calculus はプロセス代数であることから，構文，遷移規則が厳密に定義されているため，物の動作に関する性質を簡潔に，正確に表現することができる．

例として，我々が物流監視システムで扱っているプロセス式を簡略化した (1) 式を用いて説明する．

$$SEA\,[\,\cdots CY\,[CO\_a1\,[out\ CY.in\ SHIP] \quad (1)$$
$$|\,CO\_a2\,[out\ CY.in\ SHIP]\ ]\,|\,SHIP\,[]\cdots\,].$$

この式は，コンテナヤードにコンテナ a1 と a2 が存在し，コンテナヤードと並行な位置にある船に積み込もうとしている状態を表している．Ambient Calculus において，SEA や CO，$SHIP$ 等の物を表す ambient を用いて海やコンテナ，船などを表現し，また，それらを階層構造で表すことで，それぞれの位置関係を簡潔に表すことができる．また，in や out 等の $capability$ を用いることで，物の動作を表している．例えば，CO_a1 が持つ $out\ CY$ が実行されると，CO_a1 が CY の外に出て，(2) 式の形に遷移する．

$$SEA\,[\,\cdots CY\,[CO\_a2\,[out\ CY.in\ SHIP]\,|\,SHIP\,[] \quad (2)$$
$$|\,CO\_a1\,[in\ SHIP]\ ]\cdots\,].$$

3　グループ化を利用したモデル検査システム

3.1　対称性を利用したグループ化

文献 [2] の検査システムの検査方法では，最初に与えられたプロセス式に対してプロセス式の遷移グラフを作成する．遷移グラフの各ノードは，ノード ID, 親ノード ID, そしてプロセス式の構造を表す構文木を持っている．検査システムは，構文木の各頂点から実行可能な $capability$ を見つけ，遷移後の式の構文木を構成することで遷移グラフの作成を行う．可能な遷移が複数存在する場合，その動作は非決定的であるので，遷移グラフには枝分かれが生じる．例えば，(1) 式は (2) 式へ遷移可能だが，それ以外にも CO_a2 が持つ out_CY も実行可能なので，そちらが選択された場合，(3) 式の形に遷移する．

$$SEA\,[\,\cdots CY\,[CO\_a1\,[out\ CY.in\ SHIP]\,|\,SHIP\,[] \quad (3)$$
$$|\,CO\_a2\,[in\ SHIP]\ ]\cdots\,].$$

しかも海上物流ではコンテナ数は一般に数百から数千個のオーダなので，その枝分かれは膨大になり，その枝分かれに沿った検証は現実には不可能でなる．このような問題を解決するために，文献 [2] では partial order reduction と呼ばれる方法を Ambient Calculus に適用させ，遷移グラフにおいて枝刈りを行うことで，以下に示すように状態空間爆発を抑制している．

本研究で扱っている物流システムは，コンテナの数は多いが，それらの積込港と積降ろし港が同じ場合，名前をつけ換えれば同じものになる．プロセス式の中にこのような対称性が確認できれば，それらの ambient を一つの

---

* An Improvement of Ambient Logic Model Checking Algorithm for Freights
† Teppei Morita · School of Science and Engineering Kinki University
‡ Toru Kato · School of Science and Engineering Kinki University
§ Masahiro Higuchi · School of Science and Engineering Kinki University

グループにまとめる．グループ内の ambient の非決定的な動作の違いからくる複数の遷移結果は，どれか一つを代表として残し，他の遷移は遷移グラフから削除してもモデル検査には影響しない．

3.2 新たなグループ化の方法

本システムでは遷移グラフの初期ノードを構成するとき，文献 [2] のモデル検査システムと同様，構文木のどの COambient 同士が対称になるかを構造合同に基づき調べ，対称性が確認された ambient の集合を一つのグループにまとめる．本研究では (4) 式の右辺のようなプロセス式は (4) 式の左辺のような形で取り扱う．

$$\prod_{k=1}^{n} CO\_ak = CO\_a1[]|\cdots|CO\_an[] \qquad (4)$$

(4) 式の左辺から遷移を行う場合，左辺に含まれる要素から一つを任意に選択し，遷移先に同じグループに属するオブジェクトが存在しない場合，遷移先に遷移後の状態でオブジェクト化させる．既に同じグループに属するオブジェクトが存在するなら，そのオブジェクトに合流させ，新たにオブジェクトの生成は行わない．また，(4) 式中の他の要素は式のまま保持する．文献 [2] では，代表元以外でも遷移グラフのノードオブジェクトを生成し，そのノードからの遷移を考慮する際，対称性の確認を行っていたのに対し，本システムでは (4) 式を用いることで，対称性の確認を改めて行う必要がなく，代表元のみをオブジェクト化する．そのため，対称性の検査時間を大幅に短縮することができる．メモリ消費の面でも代表元以外をオブジェクト化する必要がなくなるため，本システムでは遷移グラフ中のオブジェクト数を著しく減らすことができる．これにより，COambient のオブジェクト数が，全コンテナ数からプロセス式内で並行な位置にそれぞれ一つずつになり，メモリ消費を抑えることができる．また，ambient 式をオブジェクト化する時間も短縮できる．

4 実験

```
SEA[
 KOBE[uload_v1_KOBE[in SHIP_v1] | CY[]]
 | SHIP_v1[in TOKYO.open lcomp_a1
        .out TOKYO.in KOBE]
 | TOKYO[load_v1_TOKYO[checkin(SHIP_v1) CY]
   | CY[CO_a1[checkout(load_v1)CY.in SHIP_v1
             .lcomp_a1[out CO_a1]
          | checkout(uload_v1_KOBE)SHIP_v1.in CY]]]].
```

この式は東京港から神戸港までコンテナを輸送するプロセス式である．検証にはこの式におけるコンテナ数を50〜200 とした．このプロセス式に対し，文献 [2] で用いられたシステムと，3 節で示した手法を用いて実装したシステムとを比較し，使用メモリ及び実行時間の変化を検証した．結果は表 1 のようになった．既存のシステムでは 50 個のコンテナをもつプロセス式を実行するのに 107 秒かかり，1.5GB のメモリを使用している．これに対しグループ化を行った場合，同じ 50 個のコンテナを持つプロセス式を実行すると実行時間は 10 秒，使用メモリは 860MB となっている．この結果を比較すると実行時間，使用メモリその両方の面で既存のシステムより効率的に行うことができていることがわかる．式の規模としてはコンテナ数 200 まで可能となった．

表 1　実行結果

輸送するコンテナ数	50 個	150 個	200 個
文献 [2](実行時間)	107s	73m57s	検証不能
(使用メモリ)	1.51GB	12.29GB	
本報告 (実行時間)	10s	10m13s	43m57s
(使用メモリ)	860MB	5.05GB	7.21GB

5 結論

本研究では，Ambient Calculus によりモデル化された物流システムに対するモデル検査システムのグループ化を利用した効率化，及びその検証結果について述べた．本モデル検査システムにより，プロセス式生成システムから物流書類をもとに生成されたプロセス式の検証がより効率的に行うことが可能となった．ただし，モデル検査がより効率的になったとはいえ，数千個のコンテナを含むプロセス式はまだ現実的な時間では検証できず，さらなる検査アルゴリズムが必要である．

また，単純な経路しか持たないプロセス式の検査しか行っていないので，途中で別の港に立ち寄りコンテナの積み替えを行うなどの複雑な経路を持つプロセス式でも実験を行う必要がある．

参考文献

[1] 森本大輔, 他: Ambient Calculus を用いた物流検査システム, 情報処理学会論文誌, Vol.48, No.SIG 10(PRO33), pp.151-164(2007).

[2] 加藤暢, 他: 物流システムに対する Ambient Logic 検査システム, 情報処理学会論文誌：数理モデル化と応用, Vol.3,No.1,pp.73-86(2010).

[3] Luca Cardelli and Andrew D.Gordon: Mobile Ambients, LNCS, Vol.1872, pp.1-38(2000).