

# サブクラス定義に反する閉路検知を利用した ペトリネットの詳細解析

## Detailed Analysis of Petri Nets using Closed Loop Detection violated Subclass Definition

渡貫 正也 † 張江 洋次郎 †† 和崎 克己 ††  
Masaya Watanuki Yojiro Harie Katsumi Wasaki

### 1 はじめに

ペトリネット (Petri Net) とは、事象発生の並列性、非同期性、非決定性を有する離散事象システムの振る舞いを表す数学モデルである [1]。ペトリネットの動的性質の解析における重要な概念の一つに可達性がある。可達性の解析とは、モデルが特定のトークンの配置 (マーキング) に遷移可能であるか検証することである。しかし、一般ペトリネットに適用可能な条件は現在も判明していないため、条件が判明しているサブクラスペトリネットを利用する方法 [2] が知られている。

ペトリネットの記述・解析を実現するために、本学でペトリネット援用ツール HiPS (Hierarchical Petri Net Simulator) が開発されている [3]。HiPS は、直感的で一般的な操作方法の GUI をもち、構造的・動的性質の解析機能等の様々な機能を備えている。

本研究では、可達判定条件が既知であるサブクラスの定義に反した閉路を検知する解析機能を作成する。解析結果から得られた反例を設計者に提示し、ネットの詳細解析や設計改善に利用できる。

### 2 可達条件が既知であるペトリネットサブクラス

#### 2.1 可達解析

可達性は、モデルの振る舞いを解析する上で基盤となる概念である。マーキング  $M_n$  からマーキング  $M_m$  へ遷移させるようなトランジションの発火系列が存在するとき、 $M_m$  は  $M_n$  から可達であるという。ペトリネットにおける可達解析とは、 $M_m$  が  $M_n$  から可達であるか検証することである。検証方法として主に、状態遷移図を生成する方法と、可達条件を満たすか判定する方法が存在する。

#### 2.2 トラップ、サイフォン

プレース集合  $P$  の出力トランジション集合  $P\bullet$  と、入力トランジション集合  $\bullet P$  を以下の様に定義する。

$$P\bullet = \bigcup_{p \in P} p\bullet \quad \bullet P = \bigcup_{p \in P} \bullet p$$

空でないプレース集合  $Q$  が  $Q\bullet \subseteq \bullet Q$  を満たす時、 $Q$  をトラップ (trap) と呼ぶ。また、空でないプレース集合  $S$  が  $\bullet S \subseteq S\bullet$  を満たす時、 $S$  をサイフォン (siphon) と呼ぶ。

トラップは、マーキングされた状態を維持するという

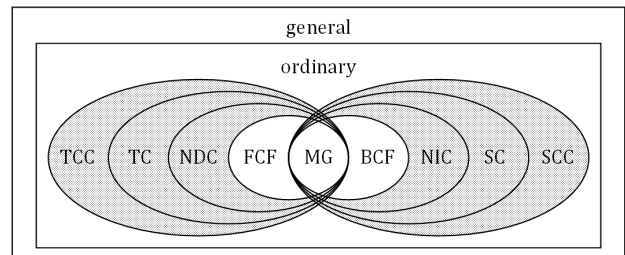


図1 可達判定条件が既知であるサブクラス群

動的性質を持つ。サイフォンは、無トークン状態を維持するという動的性質を持つ。

#### 2.3 対象サブクラス定義

ネットのアーキの重みが1である時、正規であるという。対象サブクラスネットは、すべて正規である。可達であるための必要十分条件が判明しているペトリネットサブクラス群について図1にその包含関係を示す。また、後述する極小閉路とは、構成する要素の真部分集合で閉路を構成することができない閉路を意味する。

**非減少閉路ネット (NDC) :**

すべての閉路のトークン総数が常に減少しないネット

**非増加閉路ネット (NIC) :**

すべての閉路のトークン総数が常に増加しないネット

**トラップ閉路ネット (TC) :**

すべての閉路がトラップであるネット

**サイフォン閉路ネット (SC) :**

すべての閉路がサイフォンであるネット

**トラップ包含閉路ネット (TCC) :**

すべての閉路がトラップを含むネット

(同値) すべての極小閉路がトラップであるネット

**サイフォン包含閉路ネット (SCC) :**

すべての閉路がサイフォンを含むネット

(同値) すべての極小閉路がサイフォンであるネット

### 3 サブクラス解析

解析手順のフロー図は図2である。まず、指定のサブクラスについて、ネットが包含されているか判定する。結果が偽であれば、サブクラスの定義を満たさない閉路を探索する処理に移行する。ネット上で閉路探索を実行し、探索した閉路に対してサブクラス定義に基づいた構造解析を実行する。

#### 3.1 サブクラス判定

指定されたサブクラスについて、ネットが包含されているか判定する。判定方法は、太田らの方法 [4] を利用する。TCC 判定問題は、『トラップではない極小閉路が存

† 信州大学大学院総合理工学研究所, Graduate School of Science and Technology, Shinshu University

†† 信州大学大学院総合工学系研究科, Interdisciplinary Graduate School of Science and Technology, Shinshu University

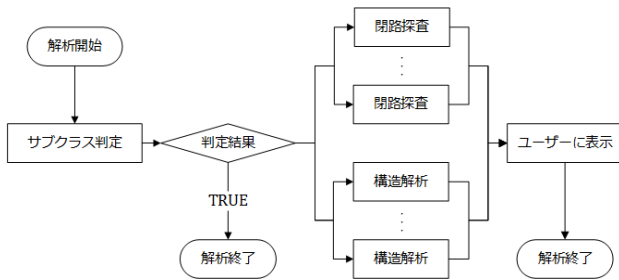


図2 サブクラス解析手順

在したとき、TCC ネットではない』を命題とした充足可能性問題に帰着させることができる。また、TC 判定問題は、『トラップではない閉路が存在したとき、TC ネットではない』を命題とした充足可能性問題に帰着させることができる。SCC 判定と SC 判定は、TCC 判定と TC 判定方法を応用することで求解できる。これらを解くためのソルバーとして、本稿では Google.OrTools[5] を用いる。

### 3.2 閉路探査

閉路探査処理には、Jhonson の基本閉路列挙アルゴリズム [6] を利用する。このアルゴリズムの計算量オーダーは、弧を  $e$ 、頂点を  $n$ 、閉路数を  $c$  として、 $O((n + e)(c + 1))$  である。

閉路探査処理は、ネットが巨大・複雑であれば膨大な計算時間がかかる。そのため、計算効率を上げるために、後述 (3.4) の機能に加えて、複数頂点から同時に閉路探査を実行する並列処理を導入する。また、機能面の充実を図り、閉路を網羅的に探査することを避けることでも、計算量の問題に対処できると考える。閉路探査機能には、探査頂点に優先度を指定する、選択したノードで探査範囲を限定するといった、閉路探査方法を詳細に指定できる機能と、随時進捗を報告した上で報告内容を参考に任意で中断できる機能を実装する。

### 3.3 構造解析

探査された閉路に対して、サブクラスの定義に適合するか解析する。解析結果から、閉路に関する動的性質の導出が可能である。ここで、閉路を構成するプレース集合を  $P_C$  とする。

NDC 構造解析では、閉路のトークン総数が非減少であるか判定する。この判定問題は、発火した時に閉路内のトークンが減少するようなトランジションが存在するという補問題として求解する。

TC 構造解析では、 $P_C$  が  $P_C \bullet \subseteq \bullet P_C$  を満たしているか判定する。

TCC 構造解析では、閉路が極小である場合、 $P_C$  が  $P_C \bullet \subseteq \bullet P_C$  を満たしているか判定する。極小でない場合は、 $P_C$  の真部分集合かつ極小閉路を構成できるプレース集合  $P'_C$  の内、 $P'_C \bullet \subseteq \bullet P'_C$  を満たしているものが存在しているか判定する。

NIC, SC, SCC 構造解析は、それぞれ NDC, TC, TCC 構造解析を応用する。

### 3.4 パイプライン型並列処理の概要

閉路探査と構造解析を非同期的に実行することで、計算の効率化と結果の早期取得を可能とする。パイプライ

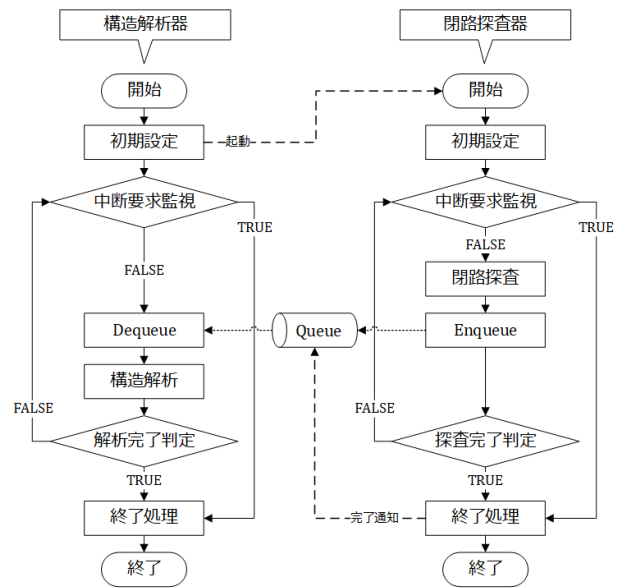


図3 パイプライン型並列処理

ン型並列処理の概念図として図3を示す。まず、構造解析器は、閉路探査器を非同期に起動する。閉路探査器は、発見した閉路を順次 Queue に格納する。構造解析器は、Queue 内を監視し、取得した閉路に対して構造解析を行う。閉路探査が終了した際は、構造解析器に Queue にこれ以上の閉路を投入しないという完了通知を行う。構造解析器は、Queue が空であり、かつ閉路探査器の完了通知を受けた後に、サブクラス判定を完了とみなし、終了する。

## 4 まとめと今後の課題

本研究では、ネットの詳細な解析や設計に利用できるツールとして、可達判定条件が既知であるサブクラス定義に反する閉路を検知する機能の作成を行った。

今後の課題は、まず、サブクラス解析機能のユーザーインターフェースを作成する。UI の作成では、閉路探査と構造解析の進捗を適切に報告する上でどのような情報を扱うか、フィードバックする解析結果の閲覧を容易とするためにどのような機能を実装すべきかという点に着目する。

そして、解析機能の性能について検証する。機能の正常動作の確認、並列処理導入による計算コストの削減量の測定、巨大・複雑なネットを含む様々なサンプルネットに対する処理時間と結果の検証等を行い、検証結果に基づいて改善点を考察する。また、機能面充実によるユーザビリティ向上について検証と改善を行う。

### 参考文献

- [1] T. Murata : Petri Nets: Properties, Analysis and Applications, Proc. of the IEEE, 77(4), 1989
- [2] 平石, 市川 : 可達性の必要十分条件を求めることが可能なペトリネットのクラス, 計測自動制御学会論文集, 24(6), 1988
- [3] HiPS : Hierarchical Petri net Simulator, <https://sourceforge.net/projects/hips-tools/>
- [4] 太田, 辻 : トラップ包含閉路ネットの検証, 電子情報通信学会技術研究報告, CAS2011-68, MSS2011-37, 2011
- [5] Google OR-Tools : <https://github.com/google/or-tools>
- [6] D. Johnson : Finding all the elementary circuits of a directed graph, SIAM J. Comput., 4(1), 1975