

7L-2

通信制御ソフトウェアの時相論理による検証 - C C I T T 勧告 X. 25 の検証 -

赤嶺 暁子 内平 直志

株式会社東芝 システム・ソフトウェア技術研究所

1. はじめに

データ通信の複雑化・多様化に伴い端末や網の相互通信における信号の送受信の形式や順序を規定するプロトコルは、ますます複雑化している。従って、これらのプロトコル仕様の論理的な誤りや意味的な誤りを検出するプロトコル検証が非常に重要になってくる。

主なプロトコル検証手法に、到達可能性解析法があり、状態グラフで記述されたプロトコル仕様に対して、定義された各プロセスの全ての状態と各プロセス間の全遷移系列を導出したグローバルな状態グラフを生成し、これを解析してプロトコルの誤りを検出する。この手法に於て問題となるのはプロトコルの複雑化に伴うグローバル状態グラフの状態数、遷移数の増大である。

本論文では、(1) 各プロセスの状態グラフを構成的検証手法により冗長な部分を縮約しながら合成し、検証の効率化を図る、(2) 論理的な誤りとセマンティクスの誤りに関する検証項目を検証項目記述言語 PQL で統一的に記述する、の二点を主な特徴とするプロトコル検証手法(システム名: 検太郎)について論じる。また、C C I T T 勧告 X. 25 のパケット通信用呼設定・解放プロトコル [1] を検証例題として検証実験を行ない、その結果に付いて触れる。

2. 検証システム『検太郎』概要

(1) 状態グラフ

『検太郎』による検証では、プロトコルで定義される各プロセスごとの仕様を状態グラフで表現する。通常、プロトコルの仕様として与えられる状態グラフは非同期式であるが、『検太郎』では各プロセス間の論理チャンネルも一つのプロセスと見なすことにより同期式の状態グラフで表現する。図1の例を考える。この例では、2つのプロセス T_1 、 T_2 が論理チャンネル C_1 、 C_2 を経由しパケット p_1 、 p_2 を送受信する。このプロトコルは、図2の論理チャンネル C_1 、 C_2 を含む4つの状態グラフでモデル化される。

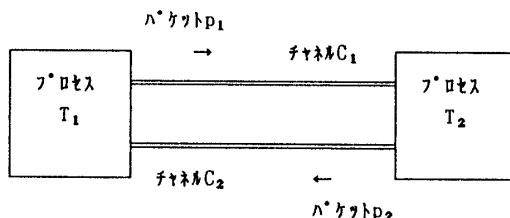


図1 プロトコル

エッジのラベルは、状態遷移に伴う動作を表す。例えば

Protocol Verification by Temporal Logic
- The X.25 Packet Level Protocol -
A. Akamine, N. Uchihiro
TOSHIBA Corporation

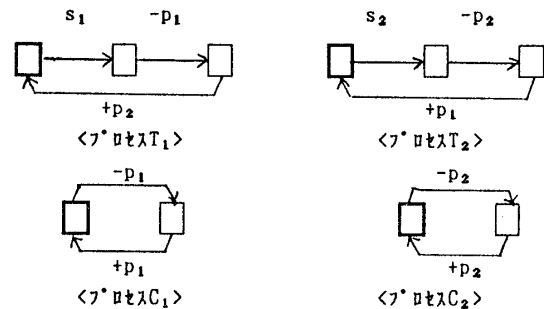
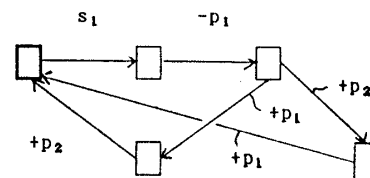


図2 状態グラフ

ば、 s_1 と s_2 は起動条件を、 $-p_1$ と $-p_2$ はパケットの送信を、 $+p_1$ と $+p_2$ はパケットの受信を表す。論理チャンネルの状態グラフは、ここでは単に、パケットが送信されたら受信されることを規定している。

さて、各プロセスの状態グラフを合成し、グローバル状態グラフを生成する。合成は、二つの状態グラフを共通の動作名でのみ同期をとりながら並行に動作させることで実現される。図3は T_1 と C_1 の合成である。

図3 $\langle T_1 \text{ と } C_1 \text{ の合成} \rangle$

(2) 検証項目記述言語 PQL

PQL (Process Query Language) は時相論理とプロセス論理を融合した言語である。『検太郎』では PQL により、CCS [2] の観測等価をベースにした非公平観測等価 [3] に基づき検証を行なう。PQL は、状態と動作の順序関係に関する高い記述能力を持ち、「デッドロックはあるか」、「未定義受信はあるか」等の論理誤りの他に、「 p_1 が送信されたら必ず p_2 が受信されるか」等の通信のセマンティクスに関する詳細な記述が可能である。

(3) 構成的検証法

プロセス T_1 と T_2 から構成されるプロトコル T を検証項目 f に関して検証する場合を考える。二つの状態グラフを合成して得られる状態グラフのサイズのオーダーは、両グラフのオーダーの和ではなく積となり状態数の爆発的な増加を生む。構成的検証法では、あらかじめ f の検証と T_1 と T_2 の同期通信に最低限必要な情報のみを保存しながら両グラフを縮約し、これを合成することにより状態数の増加を防ぐ。 f の検証に必要な情報とは f に現れる状態名と動作名であり、同期に必要な情報とは共通の動作名のことであり、これらに該当しない全ての動作名を観測不可能な内部動作 τ に書換え、 τ 選

移を非公平観測等価に基づき縮約する。図4は縮約の例である。この様にして生成された状態グラフが検証項目を充足するか否かを、グラフ上の遷移系列を辿り検出する。検証結果は充足すればYes、しなければNoが出力される。なお、縮約前と後の検証結果の等価性は保証される。

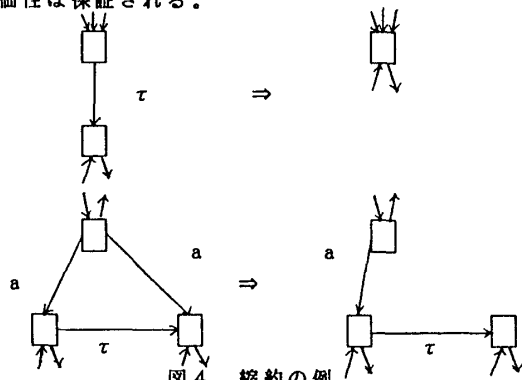


図4 縮約の例

3. X. 25 検証実験

旧X. 25は、1977年にバグが発見されている[4]。今回、PQLの検証項目記述能力と構成的検証法の効果を検討するため、このバグを『検太郎』で検証した。以下にその経過を示す。

(1) X. 25のバグ

X. 25は、DTE(端末)とDCE(網)間のパケットの送受信を規定している。旧X. 25では、論理チャネルの解放を要求する切断指示(CI)パケットが、あらゆる状態から送信可能となっており、実はこれが意味的に誤った通信の進行を引き起こす。実験では、この意味的に誤った通信の具体的な状況をPQLで記述して、これを検証する。

(2) 検証実験

A) プロセスの状態グラフ

X. 25の仕様を、DTE、DCE、送受信される8個のパケットに対応する論理チャネルを表現するプロセスの計10個のプロセスでモデル化した。各状態グラフのサイズ(遷移数)は、DTEが31、DCEが70、CIパケットが5、その他のパケットは2である。DCEとCIのサイズが大きくなるのは、CIパケットの再送条件であるエラー処理、タイムアウト処理などの仕様を情報として持たせているためである。ここでは、状態グラフの詳細については割愛する。

B) 検証項目

論理的誤りの検証項目として「デッドロックはあるか」(pq11)、意味的に誤った通信の具体的な状況を検証する項目として「CIパケットについてタイムアウトした時、DTEは既に『DCE切断指示』状態ではないのに、DCEはDTEが『DCE切断指示』状態であると解釈して、CIパケットを再送することはあるか」(pq12)の二項目をPQL表現した。検証項目は、次の論理式となる。なお、下線部は状態名と動作名、その他の部分はPQLのオペレータである。

pq11: $\text{epet internal_deadlock.}$

pq12: $\text{epet}(\text{pos}(\text{snd_CI}, \text{pos}(\text{timeout_CI},$

$\text{~dte_DCE_clear_indication \&}$
 $\text{pos}(\text{snd_CI}, \text{true}, -), -, -))$ 。

C) 構成的検証

検証項目pq11、pq12について、それぞれ構成的検証法に基づき状態グラフを生成した。pq11では、検証に要する状態・動作名は何も無く、最終的な状態グラフには動作 τ しか現われないことになる。また、pq12では、下線部の状態・動作名のみ残された状態グラフが合成される。

(3) 結果

(2)の手順に従い、『検太郎』による検証をAS4000上で行なった。表1に最終的に生成された状態グラフのエッジ数と検証結果をまとめる。case1はプロセスの状態グラフを単に順次合成した場合(縮約無し)、case2は構成的検証法の場合の結果である。case1では、グラフのサイズが大きすぎるため検証不可能であるのに対し、case2では、pq11ではエッジ数1、pq12では約64%のエッジ数のグラフが生成され、検証結果を得た。これにより、構成的検証法の効果が示されたことになる。

検証項目	エッジ数		検証結果	
	case1	case2	case1	case2
pq11	173212	1	不可	NO
pq12	173212	110298	不可	YES

表1 検証結果

4. 考察とまとめ

本章では、到達可能性解析法をベースとした検証法との比較をもとに、本検証法の特徴についてまとめる。

第一の特徴は、状態グラフの縮約法である。到達可能性解析法でも、各プロセスの状態グラフからグローバル状態グラフを導出する際の様々な縮約アルゴリズムが研究されているが、構成的検証法は、検証項目ごとに削除可能な情報をすべて縮約する点の特徴であり、これにより大きな縮約効果が期待できる。

また、一般に、到達可能性解析法では検証内容をプロトコルの論理誤りに限定し、通信のセマンティクスに関する検証は行なわない。本検証法の第二の特徴は、両者を網羅した幅広い内容の検証項目をPQLにより記述可能な点である。

本論文では、以上の二点を特徴とするプロトコル検証法とその効果について述べた。今後はこれらの特徴に関する他手法との更に詳細な比較を行なっていく。

<参考文献>

- [1] D.Belsnes, E.Lynning: Some Problems with the X.25 Packet Level Protocol, ACM SIGCOMM Comput. Commun. Rev., vol. 7, pp. 41-51, Oct. 1977.
- [2] C C I T T 勧告 X. 25
- [3] 内平直志: 様相論理による並行プログラムの検証項目記述言語PQL、日本ソフトウェア科学会第7回大会論文集、B6-2
- [4] R.Milner: Communication and Concurrency, Prentice Hall, 1989.