# 2P-03　タイマを使用した通信プロトコルの仕様記述の検証法

徳田 康平　　森 亮憲　　樋口 昌宏

大阪大学大学院基礎工学研究科

## 1　まえがき

近年，新規システムを開発する際に，システム全体を新規作成するのではなく，既に稼動しているシステムの一部を利用する形でのシステム開発が行われるようになっている [1]．ここでは，既存の機能としてタイマを利用する通信プロトコルの仕様記述の検証について考える [2]．仕様記述の検証とは，システム全体が望ましくない振舞いをしないかどうかを仕様を解析して調べることである．システムの状態はプロトコル機械の状態とタイマの状態の組で表され，仕様記述の検証はシステムが到達可能な状態を列挙することで行うことができる．しかし，タイマの設定時間が大きい場合，到達可能な状態数が膨大となり列挙の際に計算時間やメモリの点で問題となる．そこで本論文では，性質の類似した複数の状態を1つの状態集合として一括して扱い，到達可能な状態をまとめて列挙する仕様記述の検証法を提案した．以降2では想定するシステムモデルについて，3では仕様記述の検証について述べる．4では例プロトコルを用いて従来の手法と提案手法の比較を行った．

## 2　システムモデル

通信プロトコルは一般に複数のタイマを使用することができ，OSの提供するタイマ機能を用いる．図1に想定するシステムモデルを示す．

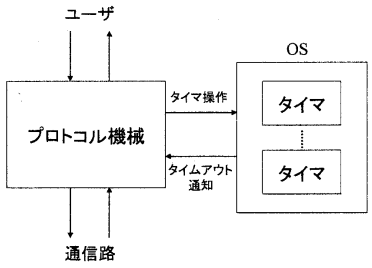


図 1: 想定するシステムモデル

### 2.1　タイマ

タイマに関する操作は，設定する (Set)，解除する (Disable)，何も行わない (Null) の3つの操作のみである．タイマはタイマごとに設定時間 (単位時間の整数倍) が決まっており，タイマが設定された後、解除されることなく設定時間が経過すると，そのタイマがタイムアウトしたことをプロトコル機械に通知する。使用するタイマは番号で識別し，設定値の小さい順に1,2,... と割り当てられるものとする．複数のタイマが同時にタイムアウトした時には，番号の若いタイマのタイムアウト通知を優先させるものとする．

### 2.2　プロトコル機械

プロトコル機械は有限状態機械で表せるものとする．プロトコル機械の各状態における入力は，外部 (ユーザまたは通信路) からの入力またはタイマからのタイムアウト通知である．出力は，外部出力と使用している各タイマに対する操作を指定するタイマ操作ベクトルである．

## 3　仕様記述の検証

### 3.1　検出する不具合

仕様記述の不具合として次のような場合を考える．

- デッドロック：
  プロトコル機械がタイムアウト通知を入力とした遷移のみ定義された状態にある時に，タイマが全て動作していない状態に陥る場合．
- 未定義のタイムアウト通知：
  プロトコル機械があるタイマ $t$ のタイムアウト時の動作が定義されていない状態にある時に，タイマ $t$ のタイムアウトが発生する．
- 実行されない遷移：
  ある状態において，あるタイマのタイムアウト通知を入力とした遷移が仕様に記述されているが，その状態でそのタイマのタイマアウト通知が発生することがない．

### 3.2　通常の方法とその問題点

システムの状態はプロトコル機械の状態 $s$ とタイマの状態 $t$ の組$< s,t >$で表される．設定時間 $N$のタイマは，$n$ 単位時間後にタイムアウトする状態 ($1 \leq n \leq N$) とタイマが動作していない状態のあわせて $(N+1)$ 個の状態で表すことができる．到達可能な状態$< s,t >$は，既に到達可能であるとわかっている状態$< s',t' >$ より，状態 $s'$で可能な遷移を解析することで求めることができる (遷移により到達可能な状態)．また，タイマの状態 $t$ に対して，全てのタイマを1単位時間減らしたタイマの状態を $t_{-1}$とした時，状

態$< s,t >$に到達可能であるなら状態$< s,t_{-1} >$にも到達可能といえる (時間経過により到達可能な状態). プロトコルの状態数, タイマの状態数共に有限であるので, 到達可能な状態$< s,t >$も有限であり, 到達可能状態集合を有限時間で求めることができる. 到達可能な状態集合が求まれば, 例えばタイムアウト通知を入力とした遷移のみが定義されている各状態$s$において, 全ての到達可能な状態$< s,t >$について$t$でいずれかのタイマが動作しているなら, デッドロックの不具合がないことが検証できる. しかしこの方法では, タイマの設定時間が大きい場合, 到達可能な状態数が膨大になるため, 計算時間やメモリの点で問題になる.

### 3.3 提案手法

従来の手法では, 到達可能な状態全てについて, 遷移により到達可能な状態を調べなければならないが, 到達可能な状態数は膨大となる場合, この作業には非常に手間がかかる. しかし, ある到達可能な状態$< s,t >$とその状態から時間経過により到達可能な状態$< s,t_{-1} >, < s,t_{-2} >, \ldots$を考えると, これらの状態はプロトコル機械の状態が同じで, 各タイマの間の大小関係も同じという類似した状態であり, これらの状態からの状態遷移を一括して扱うことができると考えられる. そこで, これら性質の類似した複数の状態を1つの状態集合として一括して扱い, 到達可能な状態をまとめて表現することを考える. 到達可能な状態の集合をプロトコル機械の状態$s$と各タイマの残り時間を表す連立不等式$R$の組$< s,R >$で表すことにする. $R$は$t_i, t_j$をタイマ$i,j$を表す変数, $C_i, C_{ij}$を定数とした時,

$$\begin{aligned} t_i &\leq C_i \\ t_i - t_j &\leq C_{ij} \end{aligned}$$

の形で表される不等式のみからなる.

ある到達可能な状態集合$< s,R >$より, 時間経過により到達可能な状態集合$< s,R' >$の$R'$は$R$中のタイマに関する不等式の定数値$C_i$を変更することにより求められる. また, ある到達可能な状態集合$< s,R >$より, 遷移により到達可能な状態集合$< s',R' >$の$R'$は$R$中の遷移の際に設定, 解除されるタイマ$t_i$に関する不等式の定数値$C_i, C_{ij}, C_{ji}$を変更することで求められる. $R'$は差分不等式で構成されているので, 解の存在判定は$O(lm)$($l$は不等式の数, $m$は変数の数)でできる [3]. よって, 連立不等式を用いて複数の状態と状態遷移を一括して扱い, 到達可能な状態集合$< s,R >$を列挙することができる. 列挙した状態集合$< s,R >$よりシステム全体で不具合が発生しないかを検証できる.

## 4 従来の手法との比較

図2の例プロトコルを用いて従来の手法と提案手法を, 列挙する状態数, 状態集合数に関して比較を行った. これは競り上げ式オークションを行うプロトコルである. タイマ1により参加者の入札間隔を監視し, 一定時間入札が行われない場合, その入札者が品物を落札する. また, タイマ2によりオークションの制限時間を監視する. 制限時間になった場合は, 最終入札者が品物を落札する.

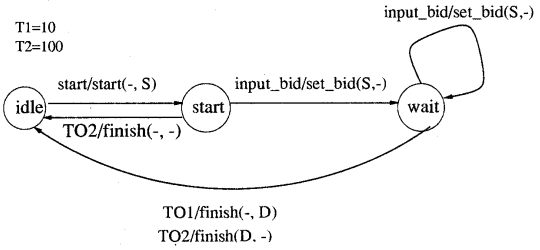


図 2: オークションを行うプロトコル

列挙する状態数の比較は, タイマ1, タイマ2の設定時間$T1, T2$をそれぞれ2,10とした時と, それぞれ10,80とした時の2通り行った. 結果は表1のようになり, 従来の手法ではタイマの設定時間が大きくなると列挙する状態数も大きくなってしまうのに比べ, 提案手法では列挙する状態集合数に変化がないことがわかる.

表 1: 列挙した到達可能な状態

| | 従来の手法 | 提案手法 |
|---|---|---|
| $(T1, T2) = (2, 10)$ | 40 | 3 |
| $(T1, T2) = (10, 80)$ | 916 | 3 |

## 5 まとめ

本論文では, タイマを既存の機能として利用する通信プロトコルの仕様記述の検証法について議論し, タイマの状態を連立不等式で表すことで複数の状態を1つの状態集合として一括して扱い, 到達可能な状態をまとめて列挙する検証法を提案した. また, 例プロトコルに適用したところ, 設定値に関わらず状態集合数を抑えることができることがわかった.

## 参考文献


[1] L.P.Lima Jr, A.R.Cavalli: A Pragmatic Approach to Generating Test Sequences for Embedded Systems, Testing of Communicating Systems, pp.288-307(1997)

[2] 森亮憲, 樋口昌宏: タイマシステムコールを用いるFSMプロトコルの適合性試験について, 情報処理学会研究報告, DPS 99-56, pp.121-126(1999)

[3] T.H.Cormen, R.L.Rivest: Introduction to Algorithms, The MIT Press, pp.539-543(1990)