ECFSMモデルの通信プロトコルへの故障耐性機能の半自動生成

2E−1

片倉 健一　樋口 昌宏　藤井 護

大阪大学 基礎工学部 情報工学科

1 はじめに

通信システムにフォールトが生じた時、システム自身でフォールトから回復できるということは重要なことである。このような機能のことをself-stabilizingと呼ぶ。しかし、大規模な通信システムの構築において、最初からself-stabilizingの機能を含むプロトコルを設計することは難しい。そこで、安全性が保証されているプロトコルを、self-stabilizingの機能を含むように拡張することは有効な方法だと思われる。多くの通信プロトコルの仕様は，拡張有限状態機械（ECFSM）としてモデル化される。ECFSMの安全性を検証する方法は、文献[1]において提案されている。また、self-stabilizingについては、文献[2]において記述されている。本稿では、フォールトが生じないという前提の下で安全性が保証されたECFSMの部分クラスであるカウンタ付FSM(FSM-C)でモデル化された通信プロトコルを基に、それをself-stabilizingの機能を含むプロトコルに半自動的に拡張する方法を提案する。

2 プロトコルモデル

本稿では通信プロトコルを2つのFSM-Cとそれらを接続する2つの無限FIFOチャネルでモデル化する。また、本モデルでは、通信路には、フォールトが生じないと仮定する。FSM-Cとは，有限状態機械(FSM)に代入，整定数の加減算，大小比較の機能を持つ整数値レジスタを付けたプロトコル機械モデルである．このようなFSM-Cを用いたプロトコルモデルの仕様の例を図1に示す．この図では，有限制御部の状態をノードで，アクションを枝で表し、2つのFSM-CをP,Qとし、2つのチャネルをPQ,QPとする。例えば、PQ:!MIP,Snは整数値Snを伴うメッセージMIPをチャネルPQに送信することを表し、QP:?MIA,Snは整数値Snを伴うメッセージMIAをチャネルQPから受信することを表す。C:NA<NS<NA+Wは遷移条件を表す。R:NS:=NS+1はレジスタ代入を表す。また、送信を伴うアクションは、まず、遷移条件を判定し、それが成り立っていれば、送信を実行し、レジスタ代入を行う。受信を伴うアクションは、受信を実行し、遷移条件を判定し、それが成り立てばレジスタ代入を行う。遷移条件が成立するアクションが存在しなければ、未定義受信となる。

Semi-Automated Generation of Fault-Tolerance Communication Protocols in ECFSM model
Kenichi KATAKURA, Masahiro HIGUCHI and Mamoru FUJII
Depadrment of Information and Computer Sciences, Faculty of Engineering Science, Osaka University
Toyonaka-shi, Osaka 560 Japan

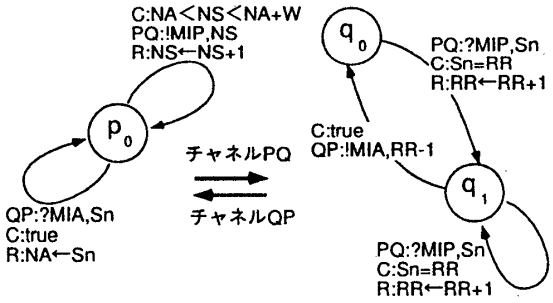

図1: プロトコルモデルの例

また、状態回復の際に文献[1]と同じ定義の不変式を用いる。不変式とはシステムが正常な実行を行う限り、常に成立する論理式のことである。不変式は積和形で表されており、本方法では特にチャネルが空であるものについて記述している積項R_iの集合を用い、項R_iは有限制御部の状態とレジスタの関係式についての表現になっている。また、FSM-Cにおいては、項R_iのレジスタの関係式は差分不等式$x-y \leq c$の積で表されているとする。ここで、x,yはレジスタを表し、cは整定数を表す。

3 プロトコルの拡張方法

3.1 デッドロック状態の発見と回復方法

ここではデッドロック状態になりうる状態を発見し、それを回復する方法を提案する。それぞれの有限制御部の状態対(p_i, q_j)に対して、図2のように状態とアクションを付け加えてデッドロックからの回復をはかる。図2において、p_i, q_jがデッドロックに陥る可能性がある有限制御部の状態であり、p_i', q_j'が新しく加える状態であり、p_k, p_lは回復後の状態である。まずプロセスPでタイマ監視を行い、タイマが切れると状態をp_i'に遷移し、Pの有限制御部の状態とレジスタの値を知らせるメッセージPiZ1..ZnをQに送信する。PiZ1..Znは有限制御部の状態がp_iであり、レジスタ$r_1,..,r_n$の値が$Z_1,..,Z_n$であることを表す。このメッセージをプロセスQが受信すると状態をq_j'に遷移し、デッドロックから回復するようにプロセスQの状態とレジスタの値を変更し、その際にPが変更すべき状態とレジスタの値を知らせるメッセージPkZ'1..Z'nをPへと送信する。プロセスPがこのメッセージを受信するとメッセージどうりに状態とレジスタの値を変更する。

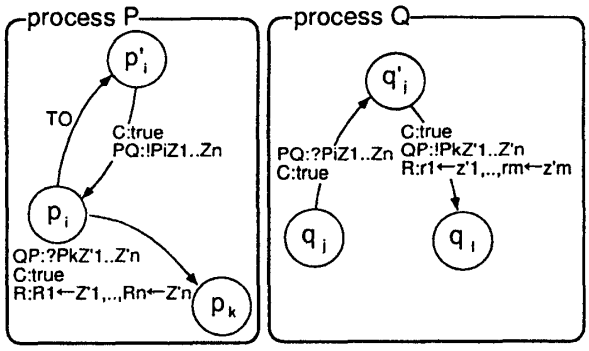

図 2: 回復方法

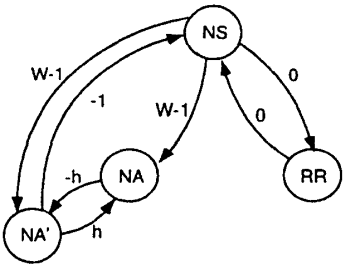

図 3: レジスタの関係式のグラフの例

3.2 回復状態を決定する方法

ここではデッドロックに陥った時に、どのように変更すれば回復できるかを説明する。まず、デッドロック状態（有限制御部とレジスタの値における）と不変式の空チャネルについての積項 R_i の距離 Dを定義する。はじめにデッドロック状態にある有限制御部の状態を何個変更すれば項 R_i の状態になるかを、有限制御部の状態における距離 D_sとする。次にデッドロック状態にあるレジスタの値を何個変更すれば R_i のレジスタの関係式を満たすようになるかを、レジスタにおける距離 D_rとする。このとき距離 D を D_sと D_rの和とする。デッドロック状態とすべての項 R_i の距離 Dを調べ、その中でもっとも Dの値が小さくなるような項 R_{min}を探し、その R_{min}を満たすように有限制御部の状態とレジスタの値を変更して、システムを回復する。ここで、距離 D_sは状態を比較して調べることができる。そこで、次に距離 D_rをどのように調べるかを説明する。また、例えば変更するレジスタの値は、デッドロック状態にあるレジスタの値から項 R_{min}のレジスタの関係式を満たすもっとも近い値にすればよい。

3.2.1 レジスタにおける距離を調べる方法

ここでは文献 [3] の連立不等式が解を持つかの判定問題をグラフの最短路問題を用いて解く方法を利用する。有向グラフ $G = (V, E)$ を、レジスタの名前の集合を Vとし、不等式 $x - y \leq c$ を表すノード x からノード yへ の c と重みをつけた有向辺の集合を Eとする。G の中に重みの和が負となる閉路がなければ、すべてのレジスタの不等式を満たすレジスタの値が存在するということである。本方法では、あるレジスタを変更した時、項 R_iのすべてのレジスタの不等式を満たすかを調べることにより、レジスタにおける距離 D_rを決定する。以下にこの方法の手順を示す。

1. 項 R_iのレジスタの不等式の中で、デッドロック状態にあるレジスタの値で満たされているものを E に加える。項 R_iのレジスタの不等式の中で、デッドロック状態にあるレジスタの値で満たされていないものを $C_1, .., C_n$とする。

2. ある x_iについて、x_iがすべての $C_1, .., C_n$に現れるならば、$x_i' = x_i + h$ とし、x_i'を Vに加えたものを V'とし、不等式 $x_i' - x_i \leq h_i, x_i - x_i' \leq -h_i$と、項 R_iのすべてのレジスタの不等式において変更するレジスタ x_iを x_i'に置き換えた不等式を Eに加えたものを E'とする。グラフ $G' = (V', E')$ 中のすべての閉路の和を非負として連立不等式にし、h_iが 0以外の値を持つ場合、D_r は 1である。

3. 上記のような x_iがないか、h_iが 0以外の値を持たない場合、ある x_{i_1}, x_{i_2}について、x_{i_1}, x_{i_2}のどちらかがすべての $C_1, .., C_n$に現れるならば、$x_{i_1}' = x_{i_1} + h_{i_1}, x_{i_2}' = x_{i_2} + h_{i_2}$とし、$x_{i_1}, x_{i_2}$を Vに加えたものを V''とし、上記と同様に不等式を Eに加えたものを E''とする。グラフ $G'' = (V'', E'')$ 中のすべての閉路の和を非負として連立不等式にし、h_{i_1}, h_{i_2}の両方が 0以外の値を持つ場合、D_r は 2である。

4. 上記のような x_{i_1}, x_{i_2}がないか、h_{i_1}, h_{i_2}が 0以外の値を持たない場合、D_rは 3以上である。

3.2.2 例

ここでは、図 1 のプロトコルにおいての項 R_iのレジスタの関係式 NS-RR=0 ∧ NA-NS ≤ -1 ∧ NS-NA ≤ W-1 を対象とし、デッドロック状態において、NA-NS ≤ -1 が成り立っていないとする。上記の方法により、レジスタ NA を変更し、図 3 のレジスタの関係式のグラフを得ることができる。このとき、閉路のラベルの総和に h が現れるのは $-h$+W-2 だけである。これを正としても h の値は存在するので、レジスタ距離 D_rは 1である。

4 おわりに

本稿では、デッドロック状態をタイマ監視により検出し、それを回復する方法について説明したが、未定義受信によるフォールトも少しの変更を加えることにより対応できると考えており、今後拡張していく予定である。

参考文献

[1] Masahiro Higuchi ,et al.: A Verification Method via Invariant for Communication Protocols Modeled as Extended Communicating Finite-State Machines, *IEICE Trans.Commun.*,vol E76-B,No 11,pp.1363-1372,November 1993.

[2] Anish Arora ,et al.: Closure and Convergence:A Foundation of Fault-Tolerant Computing, *IEEE Trans. Software Eng.*, vol 19,No 11,pp.1015-1027,November 1993.

[3] Cormen,T.H.,et al.: Introduction to Algorithms, *The MIT Press*, pp.539-543,1990.