

グラフ変換を用いたプロトコル合成法

田倉 昭

世良孝文

太田理

A T R 通信システム研究所

〒 619-02 京都府相楽郡精華町光台二丁目二番地

Tel: 0774-95-1236 E-mail: takura@atr-sw.atr.co.jp

あ ら ま し 複数端末の大域状態遷移規則の集合として記述された通信サービス仕様から端末毎に局所状態を保持するプロセスの仕様を導出するプロトコル合成法を提案する。通信サービス仕様を有向グラフの置き換え規則の集合として表現し、適用条件を満足する規則探索のためのネットワーク上の通信を行うプロトコルエンティティの仕様を導出する。各プロトコルエンティティは、通信システム全体を表す有向グラフに含まれる同型な部分グラフを探索する分散アルゴリズムを実現する。一つの通信サービスにかかわる端末の数には制限があるので、実用的な時間で動作する仕様を導出することができる。生成されたプロトコル仕様は最終的に通信システム上で動作可能なプログラムに変換することができ、実際の通信ソフトウェア開発に適用することが可能である。

キーワード 仕様記述, グラフ変換, プロトコル合成, 通信ソフトウェア, 通信サービス

Protocol Synthesis Using Graph Replacement

Akira TAKURA

Takafumi SERA

Tadashi OHTA

ATR Communication Systems Research Laboratories

2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-02, Japan

Tel: +81-774-95-1236 E-mail: takura@atr-sw.atr.co.jp

Abstract A protocol synthesis method from communications service specifications is proposed for implementing a distributed graph replacement algorithm. Communications service specifications are represented by a set of graph replacement rules. Generated protocol entity specifications are implemented to find and apply the proper rule for using communications to traverse protocol entities in a network. The generated protocol entity specifications operate within a time practical for application because of the limited number of terminals associating a communications service. The generated protocol entity specifications can be refined to a program implementing specified services. Accordingly, the proposed protocol synthesis method can be used to develop communications software.

key words Specification, graph replacement, protocol synthesis, communications software, communications service

1 まえがき

マルチメディアサービスをはじめとして通信サービス開発に対する要求は増大しつつある。更に、通信ネットワークはユーザが個別のサービスを開発できるように発展しつつある [1]。形式仕様記述言語に基づく通信ソフトウェアの自動生成は、このような要求に応えることができる有効な手法である。そこで、通信システムの構造や通信サービスの実現法を知らない通信システムの非専門家が記述可能な通信サービス仕様記述言語を用いて書かれた通信サービス仕様から通信ソフトウェアを自動生成する方法を提案する。本論文では、通信ソフトウェア自動生成の一段階として、サービス仕様からプロトコルを自動合成する方法を提案する。プロトコル仕様から通信システム上で動作するソフトウェアへの変換は従来提案されている手法 [2] を用いて行うことができる。

通信サービスを対象とした仕様記述言語として、SDL [3] や LOTOS [4] を始めとして多くの言語が提案されている。SDL は通信システムのソフトウェア開発言語として広く使われている [5]。LOTOS もまた通信サービス仕様の形式的な記述言語として使うことが可能である [6, 7]。これらの仕様記述言語を通信ソフトウェア開発に適用する場合、仕様からソフトウェアを得るために段階的な詳細化が必要である [8, 9, 10]。例えば、Cameron 他 [8] は、規則ベースの仕様記述言語 L.0 を用いて実際のプロトコルを実現している。Tsai 他 [9] は、フレームと規則ベースの要求記述言語 FRORL を用いている。これらの方法では、仕様を段階的に詳細化してソフトウェアを導出している。この過程で、プログラム内部の動作を理解する必要がある。非専門家を対象としたソフトウェア開発には無理がある。通信ソフトウェアの自動生成を目指して、Eijk 他 [11] は LOTOS から C プログラムへの変換を行っている。この場合でも、途中の過程で人間の介入が必要である。

Bochmann, Gotzhein [12, 13], Chu 他 [14], Saleh 他 [15], 角田他 [16] は、サービス仕様からのプロトコル合成法を提案している。これらの方法では、通信システムへのアクセス点におけるすべての動作を規定する。本論文では、通信システムの非専門家を対象とするソフトウェア自動生成を対象とするため、サービス仕様の記述言語として、分散システムである通信システムで起こる動作のうち通信システムの外部から認識可能な動作のみで仕様を記述する言語 STR [17] を用いる。STR はプロダクション規則の集合で通信サービス仕様を定義する。具体的には、通信システムの外部から認識可能な動作として通信システムに接続

されている端末の動作を宣言型規則の集合で規定する。通信サービスは複数の端末の複合動作として規定されるので、サービス仕様は複数端末のグローバルな状態遷移となる。通信システムは地理的に分散した分散システムであるので、グローバルなサービス仕様記述から各端末毎のローカルな状態に関する情報に基づいて必要な通信を行ってサービスを実現するソフトウェアを自動生成する。

通信システムに接続されている端末の状態は有向グラフとして表現することができる。このグラフ表現を用いると、通信サービスは端末で起こったイベントを起点とするグラフの置き換え規則の集合とみなすことができる。グラフの置き換え規則として表現された通信サービス仕様から、そのサービス仕様を実現するソフトウェア仕様を導出する手法を提案する。従来、適用条件を表すグラフに分岐がないという制限が設けられていた [18] が、本論文では仮想的に分岐のないグラフに変換することにより、任意のサービス仕様からのソフトウェア自動生成を行うことが可能である。

2 通信サービス仕様とプロトコル合成

2.1 通信ソフトウェアモデル

サービス仕様からプロトコルを合成することにより、サービス仕様を満足する誤りのないプロトコルを自動生成することができる。従来、提案されているプロトコル合成法は、図 1 に示す OSI 参照モデルに代表される階層型アーキテクチャに基づいている。レイヤ (N) はレイヤ (N-1) から提供される (N-1) サービスを利用し、通信相手のレイヤ (N) と相互通信を行って、レイヤ (N+1) に提供する (N) サービスを実現する。レイヤ (N) 内のエンティティ間で通信を行うための通信規約を (N) プロトコルと呼ぶ。レイヤ (N+1) がレイヤ (N) サービスにアクセスする点を (N) SAP と呼ぶ。各 SAP 対応に一つのプロトコルエンティティが対応する。以下では、プロトコルエンティティのことをプロセスと呼ぶことがある。電話サービスをはじめとする通信サービスでは、端末が図 1 における SAP に相当する。

2.2 サービス仕様とプロトコル仕様

Bochmann, Gotzhein [12, 13], Chu 他 [14], Saleh 他 [15] は、図 1 に示されるサービスとプロトコルの関係を仮定して、サービス仕様からのプロトコル合成法を提案している。また、角田他 [16] は、複数の SAP で同時にイベントが生起する場合を含むプロトコル合成法を提案している。通信システムは、SAP1, ..., SAPn を通してシステムに

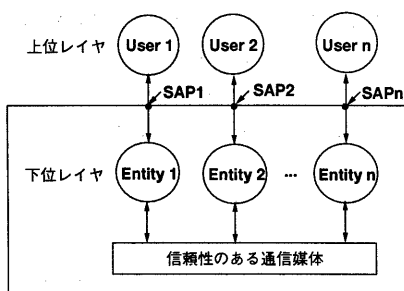


図 1: 階層型プロトコルアーキテクチャ

アクセスするユーザに通信サービスを提供する。このモデルにおいて、サービス仕様とプロトコル仕様は以下の通り定義される [19]。

- サービス仕様は下位プロトコル階層がその上位プロトコル階層に提供するサービスを表す。下位階層によって提供されるサービスは、SAP における動作を規定するサービスプリミティブを用いて定義される。
- プロトコル仕様は下位階層のプロトコルエンティティ間の相互作用を規定する。この相互作用は、上位階層に提供されるサービスと通信媒体から提供されるサービスとによって定義される。

3 通信サービス仕様記述

3.1 STR

通信サービスは状態遷移機械としてモデル化することができる [20]。通信サービス仕様を通信システムの外部から認識可能な端末の状態遷移規則の集合により規定する通信サービス仕様記述言語 STR の定義を述べる。

端末の状態を、端末の状態を表現する状態記述要素の集合で定義する。状態記述要素は一つまたは二つの引数を持つ。第一引数で指定される端末が、その状態記述要素で指定される状態にあることを表す。第二引数が記述されている場合には、その状態記述要素は第一引数の端末が第二引数の端末とその状態記述要素で表される関係にあることを意味する。すべての端末は初期状態として空き (idle) 状態を持つ。

端末で生じたイベント (端末を介した通信システムへの入力) に対する端末の状態変化を規則として表現する。一つの規則は次の形で書かれる。

$$p_1, \dots, p_m \quad e: \quad q_1, \dots, q_n.$$

上の規則において、 $p_1, \dots, p_m$ を現状態、 e をイベント、 $q_1, \dots, q_n$ を次状態と呼ぶ。現状態、次状態の構成要素である $p_i (1 \leq i \leq m)$, $q_j (1 \leq j \leq n)$ を状態記述要素と呼ぶ。

イベントは一つあるいは二つの引数を有し、第一引数で指定される端末においてそのイベントが発生したことを表す。イベントの第二引数は、そのイベントにおいて入力された他端末の識別情報を表す。上記の規則はイベントが発生したときに、そのイベントにより端末の状態変化が起こる端末群の状態が現状態から次状態に移移することを表す。一つの規則内で同じ端末変数は同一の端末を表す。

3.2 STR のグラフ表現

通信システムに属するすべての端末の状態を次の対応関係により一つの有向グラフ (連結とは限らない) として表現することができる。このグラフをシステムグラフと呼ぶ。

- 端末を頂点、1 引数の状態記述要素及びイベントを頂点のラベル、2 引数の状態記述要素及びイベントを第一引数から第二引数へのラベル付き有向辺として表現する。イベントの第一引数に対応する頂点をイベント生起点と呼ぶ。

上の対応関係を用いると、一つの STR 規則は、現状態とイベントのグラフ表現である始グラフと次状態のグラフ表現である次グラフからなる二つの有向グラフの組として表現することができる。このグラフ表現を用いると STR 規則の適用は、条件を満足する部分グラフを探索し、適用すべき STR 規則の次グラフで置き換える問題として定式化することができる。

3.3 STR 規則の適用法

端末の状態、STR 規則の適用法を定義する。 t を端末、状態記述要素 $p_1, \dots, p_l$ の第一引数が t であると仮定する。このとき、 t がもつ状態記述要素が $p_1, \dots, p_l$ であるとき、 t は $p_i (i = 1, \dots, l)$ を満足する、あるいは t は状態 $p_i (i = 1, \dots, l)$ にあると定義する。

ある規則 r がある端末群 $t_1, \dots, t_n$ に対して適用できる条件は、 r に含まれるイベントの発生端末が $t_1, \dots, t_n$ のどれかであり、規則 r の現状態にある状態記述要素がすべて $t_1, \dots, t_n$ により満足されるときである。これは、 r の

始グラフがシステムグラフの部分グラフとして含まれることを表す。

複数の STR 規則が与えられたとき、上記の適用条件を満足する規則は必ずしも一つに定まるとは限らない。二つの STR 規則 r_1 , r_2 がともに適用条件を満足し、それぞれのグラフ表現において、 r_2 の始グラフが r_1 の始グラフのラベルを含めた部分グラフ ($r_2 < r_1$ と書く) となるとき、 r_1 のほうが r_2 より適用の優先順位が高いと定義する。

ある端末の集合とそれらの状態が与えられたとする。このとき、STR 規則の適用条件を満足する複数の STR 規則があるとき、それらの中で適用の優先順位が最も高いものを適用する。最も優先順位が高い STR 規則が一つに定まらない場合には、優先順位の高い STR 規則が存在しない STR 規則の中から任意の一つを適用する。

ある端末の集合と適用すべき STR 規則 r が決定したとき、 r の始状態で規定されている状態記述要素を r の次状態で規定されている状態記述要素で置き換えることを r の適用と定義する。図 2 に、STR 規則が適用される状況を图示する。

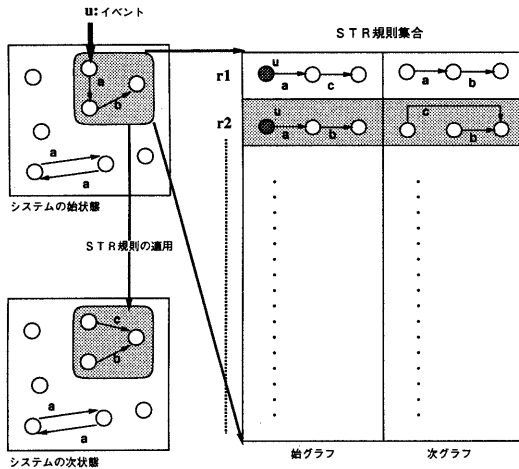


図 2: STR 規則の適用

4 プロトコル合成

4.1 準備

STR で記述されたサービス仕様からのプロトコル合成法が必要となる用語の定義を行う。

イベントが e である規則の始グラフの集合を $C(e)$ と表す。

規則包含関係グラフ $C(e)$ に含まれる複数の要素に共通し、イベント生起点を含む部分グラフを s とし、 s を部分グラフとする $C(e)$ の要素の集合を $D(s)$ とおく。 s に辺を追加して得られるグラフ s' が $C(e)$ の要素の部分グラフであり、 $D(s) \neq D(s')$ であるとき、 $\langle s, D(s) \rangle$ は規則包含関係グラフの頂点である。 $r \in C(e)$ ならば、 $\langle r, D(r) \rangle$ は規則包含関係グラフの頂点である。規則包含関係グラフの二つの頂点 $\langle s, D(s) \rangle$ と $\langle t, D(t) \rangle$ の間に、(1) $s < t$ であり、(2) $s < u$, かつ $u < t$ である他の要素 $\langle u, D(u) \rangle$ が存在しないとき、 $\langle s, D(s) \rangle$ から $\langle t, D(t) \rangle$ への有向辺が存在する。このとき、 $\langle s, D(s) \rangle$ は $\langle t, D(t) \rangle$ より小さいと定義する。

図 3 に示す始グラフに対する規則包含関係グラフを図 4 に示す。

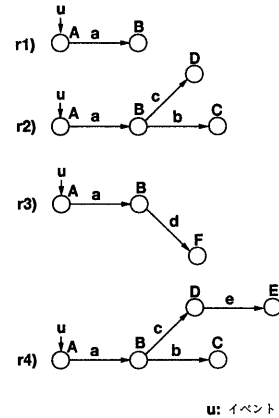


図 3: 始グラフの例

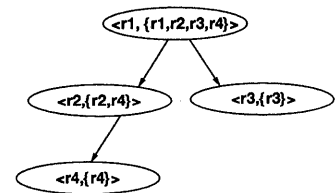


図 4: 規則包含関係グラフの例

規則被覆木 規則 r のグラフ表現を $g_r = (v_r, e_r)$ とする。 g_r の一つの頂点被覆木を $t_r = (w_r, f_r)$ とする。 g_r の規

則被覆木 c_r を次の通り定義する。

c_r の頂点の集合は、 v_r と一致する。

規則被覆木の同一頂点から出る複数の辺がある場合には、辺についているラベルの順序により時計回りに並べる。 c_r の辺の集合は、 e_r と次の二つの条件を満足する辺を追加したものと一致する。

1. t_r の葉に当たる頂点にラベルがある場合には、 t_r の葉に対応する c_r の頂点を始点とする辺を c_r の辺に追加する。
2. c_r の辺 ab は、2 項組からなるラベルをもつ。ラベルの第一要素は、 a に対応する v_r のラベルであり、ラベルの第二要素は、 e_r における辺 ab のラベルである。

この規則被覆木は、規則グラフの頂点被覆木に対して唯一に定まる。次に定義する規則被覆重ね合わせ木に用いる規則被覆木は、次の二つの条件を満足する頂点被覆木から生成される規則被覆木とする。規則 r に対する頂点被覆木を s 、 s から生成される規則被覆木を t とする。

- (1) s は r に対する規則グラフ g_r のラベルなし部分グラフである。
- (2) s の一つの頂点を u とする。 g_r を一つの頂点とする規則包含関係グラフ G において、 G の二つの頂点 g, h の間に、 $g < h$ の関係があるとすると、このとき、 u が h の頂点であり、 u が g の頂点ではない場合には、 g のある頂点から u に至る辺が s に含まれる。

図5に図3の始グラフに対する規則被覆木を示す。

規則被覆重ね合わせ木 次の条件を満足するラベル付き有向グラフを規則集合 $C(e)$ に対する規則被覆重ね合わせ木と呼ぶ。

1. $C(e)$ の各要素に対して、一つの規則被覆木が部分グラフとして含まれる、
2. 二つの規則被覆木のある頂点が重ね合わされているとき、それらの頂点から出ている二つの辺を時計回りに、等しいあるいは包含関係のあるラベルをもつ辺どうしを重ね合わせる。

仮通信経路 規則被覆重ね合わせ木におけるイベント生起点を出発点として、すべての頂点をちょうど一回だけ通過し、規則被覆重ね合わせ木の二つの頂点を u, v としたとき、次のいずれかの条件を満足するときには仮通信経路において u は v より前にある。

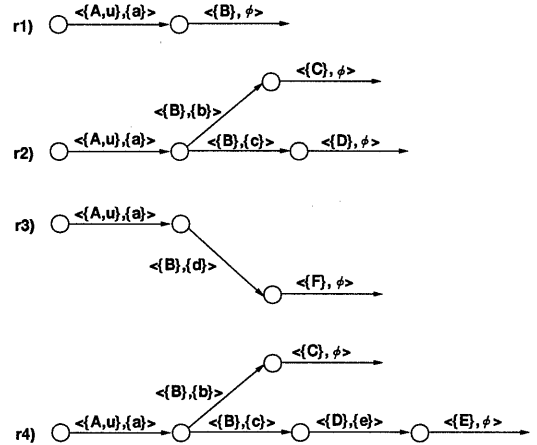


図 5: 規則被覆木の例

1. g, h を $C(e)$ に対する規則包含関係グラフの二つの頂点を構成するグラフであるとする。 $g < h$ 、かつ u は g, h の頂点であり、 v は g の頂点ではないが h の頂点である。
2. h を $C(e)$ に対する規則包含関係グラフの頂点を構成するグラフであるとする。 $g < h$ である $C(e)$ に対する規則包含関係グラフの任意の頂点 g に対して、 u, v が h の頂点であり、 g の頂点でないとする。 h において、 u がイベント生起点と v を結ぶ経路上にある頂点か、またはイベント生起点と u を結ぶ経路上の頂点 w があり、 w がイベント生起点と v を結ぶ経路上の頂点であり、かつ w は g の頂点ではない。

図6に図3の始グラフの集合に対する、規則被覆重ね合わせ木およびそれに対する仮通信経路を示す。

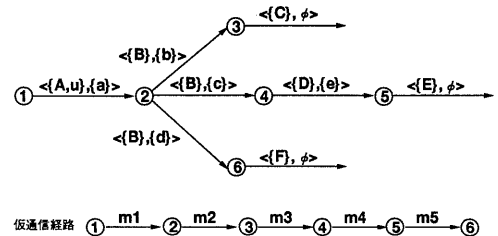


図 6: 規則被覆重ね合わせ木の例

接続関係判定 仮通信経路に沿った通信において、仮通信経

路上の辺と一致しない辺の接続関係の判定を行う処理を表す。仮通信経路上の二つの頂点を i, j とし、 i は j より仮通信経路において根に近いとする。このとき、 i を始点とする辺のうち、仮通信経路と一致しない辺の接続先のうち、 j または j を始点とする辺のうち仮通信経路と一致しない辺の接続先が一致するかどうかの判定を行う処理を表す。

図7に接続関係判定が必要な二つの始グラフの例を示す。図の二つの始グラフはどちらも1, 2, 3が仮通信経路である。プロセス1, 3だけがもつ情報だけで個別に判定可能な範囲は図8に示す部分グラフである。プロセス1とプロセス3から出ている辺 b, d の先が異なるか等しいかで区別される図7の二つの始グラフは、プロセス1から出ている辺 b の先が4であることをプロセス3が受けとって、プロセス3ではじめて判定することができる。

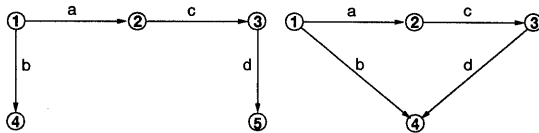


図 7: 接続関係判定の必要な二つの始グラフ

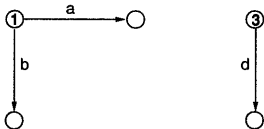


図 8: 局所状態だけで判定可能な範囲

探索信号 仮通信経路上の隣接頂点間の辺に唯一の名前をもつ信号である。合成されたプロトコルにおいては、信号名の他に以下の情報が含まれる。

仮確定規則：今までに通過してきたプロセスの適用条件判定により、適用条件が満足されていることが確定している規則のうち、もっとも適用優先順位の高い規則。

規則候補集合：適用条件判定が完了していない規則の集合。

接続関係情報：接続関係判定に必要なプロセス情報。

通過プロセス列：イベント生起後に通過してきたプロセスの系列。

仮確定規則プロセス列：仮確定規則に対応するプロセスの系列。

応答信号 応答信号には適用すべき規則名と適用すべき規則がないことを表す信号の二種類がある。適用規則名を表す信号の場合には、その応答信号を受けとるプロセスが仮通信経路上の何番目のプロセスに対応するかを表す情報を含む。

状態遷移片 各規則の始グラフにおける頂点のうち、ラベルがあるかその頂点を始点とする有向辺がある頂点毎に、一つあるいは二つの状態遷移片が対応する。

- ・頂点が仮通信経路上で、その規則の最後に出現する頂点である場合には、仮通信経路上で確定している探索信号を受けとった後、確定した規則を表す応答信号を送信し、規則で決定される次状態に移移する状態遷移片が対応する。

- ・頂点が仮通信経路上で、その規則の終端でない場合には、仮通信経路上で確定している探索信号を受けとった後、規則の適用条件を仮通信経路に沿って、更に判定する探索信号を送信し、元の状態に戻る状態遷移片と、確定した規則を表す応答信号を受けとり、規則で決定される次状態に移移する状態遷移片の二つが対応する。

図3の4つの始グラフの頂点Bに対して、図6で決定された仮通信経路上の信号を用いて作成した状態遷移片を図9に示す。なお、図では探索信号を受けとった直後に行う接続関係判定を省略してある。

4.2 プロトコル合成法

以下の手順により STR で記述されたサービス仕様からプロトコルを合成する。

(1) イベント毎の規則グラフの分類

同一イベントを持つ規則グラフ毎に分類する。イベント e に対する分類を $C(e)$ とおく。得られた分類毎に以下の2～5を行った後で、6を実行する。

(2) 規則包含関係グラフの作成

規則包含関係グラフを作成する。

(3) 規則被覆重ね合わせ木の作成

規則被覆重ね合わせ木を作成する。

(4) 仮通信経路、探索信号の決定

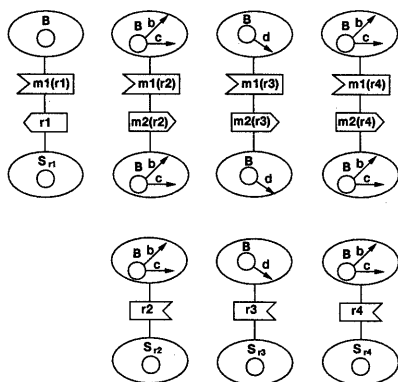


図 9: 状態遷移片の例

規則被覆重ね合わせ木をもとに、仮通信経路、探索信号名を決定する。

(5) 状態遷移片の作成

仮通信経路をもとに決定した信号名を使って、各規則 r に対応する規則被覆木の頂点毎に、状態遷移片を作成する。始状態は、規則被覆木の頂点から出ている辺に付加されているラベルの第一要素である。

(6) プロセス仕様の合成

初期状態から順に状態遷移片を合成し、新たな状態が生成されなくなるまで、合成を繰り返し、最終的に全体のプロセス仕様を求める。合成の過程で、ある新しい状態 s が合成されたとする。このとき、 s の部分グラフとして始状態が含まれる状態遷移片をすべて集める。集めた状態遷移片に対して次の方法で合成を行う。

- 適用条件判定が終了する規則の状態遷移片と更に適用条件判定が必要な規則の状態遷移片がともに含まれる場合

適用条件判定が終了する規則の状態遷移片の次状態を始状態と同じにして状態遷移片の合成を行う。状態遷移片の合成において、受信した探索信号に含まれる規則候補集合と合成対象となった状態遷移片に対応する規則集合との共通集合が他プロセスへの探索信号となる。

この共通集合が空でなければ、共通集合に含まれる規則から仮通信経路上の次に探索信号を送るべきプロセスを決定し、探索信号を送る。共通集合が空の場合には、仮確定規則がある場合には、その規則が適用規則となり、仮確定規則がない場合には、非遷

移指示信号をすべての通過プロセスに送る。

- 適用条件判定が終了する規則の状態遷移片のみ、または更に適用条件判定が必要な規則の状態遷移片のみの場合

集められた状態遷移片をそのまま合成する。状態遷移片の合成のしかたは上記と同じである。

合成されたプロセス仕様は、通信が進むにつれて適用すべき規則が絞られていき、最終的に適用すべき規則が決まる。適用すべき規則が決定した時点で、該当するプロセスがイベント生起後、通過してきたプロセスに応答信号を送る。応答信号には、遷移指示信号と非遷移指示信号がある。遷移指示信号を受けとったプロセスは、状態遷移片で指定された状態に遷移する。非遷移指示信号を受けとったプロセスは、探索信号受信前の状態に遷移する。

5 結論

グラフの置き換え規則として表現される通信サービス仕様から、それを実現するプロトコルを合成するアルゴリズムを示した。生成されるプロトコルは、分散システム上で、グラフの同型判定を行う分散アルゴリズムを実現することになるが、実用上は問題がない。すなわち、電話サービスを始めとする通常の通信サービスでは、一つの呼にかかわる端末の数はそれほど多くはなく、組合せ的な爆発はほとんど起こらない。

提案したプロトコル合成法を用いて、通信システムをブラックボックスとみなした通信サービス仕様からソフトウェアを自動生成することができる。この結果、通信ソフトウェア開発への非専門家の参加が可能となる。

謝辞 本研究を進めるにあたり、熱心な御討論を頂いた ATR 通信システム研究所通信ソフトウェア研究室の皆様に感謝いたします。

参考文献

- [1] Garrahan, J. J., Russo, P. A., Kitami, K. and Kung, R., "Intelligent Network Overview," *IEEE Commun. Mag.*, pp. 30-36, Mar. 1993.
- [2] Takura, A., Kawata, K., Ohta, T. and Terashima, N., "Communication Software Generation Based on Two-Layered Specifications and Execution Environment," in *IEEE GLOBECOM'93* (Houston, USA), pp. 362-368, Dec. 1993.

- [3] CCITT, revised Recommendation Z.100, "CCITT Specification and Description Language (SDL)," May 1992.
- [4] Bolognesi, T. and Brinksma, E., "Introduction to the ISO Specification Language LOTOS," *Comput. Networks ISDN Syst.*, vol. 14, pp. 25-59, 1987.
- [5] Belina, F. and Hogrefe, D., "The CCITT-Specification and Description Language SDL," *Comput. Networks ISDN Syst.*, vol. 16, pp. 311-341, 1989.
- [6] Faci, M., Logrippo, L. and Stepien, B., "Formal Specification of Telephone Systems in LOTOS: the Constraint-Oriented Style Approach," *Comput. Networks ISDN Syst.*, vol. 21, pp. 53-67, 1991.
- [7] Drayton, L., Chetwynd, A. and Blair, G., "Introduction to LOTOS through a worked example," *Comput. Commun.*, vol. 15, no. 2, pp. 70-85, Mar. 1992.
- [8] Cameron, E. J., Cohen, D. M., Guithner, T. M., Keese Jr., W. M., Ness, L. A., Norman, C. and Srinidhi, H. N., "The L.O Language and Environment for Protocol Simulation and Prototyping," *IEEE Trans. on Comput.*, vol. 40, no. 4, pp. 562-571, Apr. 1991.
- [9] Tsai, J. J. P., Weigert, T., Jang, H.-C., "A Hybrid Knowledge Representation as a Basis of Requirement Specification and Specification Analysis," *IEEE Trans. on Software Eng.*, vol. 18, no. 12, pp. 1076 - 1100, Dec. 1992.
- [10] Dendorfer, C. and Weber, R., "From Service Specification to Protocol Entity Implementation - An Exercise in Formal Protocol Development," in *Proc. IFIP Twelfth Int. Symp. Protocol Specification, Testing, Verification*, pp. 163-177, 1992.
- [11] Eijk, P. v., Kremer, H. and Sinderen, M. v., "On the use of specification styles for automated protocol implementation from LOTOS to C," in *Proc. IFIP Tenth Int. Symp. Protocol Specification, Testing, Verification*, pp. 157-168, 1990.
- [12] Bochmann, G. v. and Gotzhein, R., "Deriving protocol specifications from service specifications," in *Communications, Architectures & Protocols, Proc. ACM SIGCOMM '86* (Vermont, USA), pp. 136-145, Aug. 1986.
- [13] Gotzhein, R. and Bochmann, G. v., "Deriving Protocol Specifications from Service Specifications Including Parameters," *ACM Trans. Comput. Systems*, vol. 8, no. 4, pp. 255-283, Nov. 1990.
- [14] Chu, P. M. and Liu, M. T., "Synthesizing Protocol Specifications from service specification in the FSM model," in *Proc. Comput. Networking Symp.*, pp. 505-512, Apr. 1988.
- [15] Saleh, K. and Probert, R. L., "A Service-Based Method for the Synthesis of communications protocols," *Int. J. Mini and Microcomput. Special Issue on Distributed Systems*, vol. 12, no. 3, pp. 97-103, 1990.
- [16] Kakuda, Y., Nakamura, M. and Kikuno, T., "Automated synthesis of protocol specifications from service specifications with parallelly executable multiple primitives," *IEICE Trans. Fundamentals*, vol. E77-A, no. 10, Oct. 1994.
- [17] Hirakawa, Y. and Takenaka, T., "Telecommunication Service Description Using State Transition Rules," in *Proc. Sixth Int. Workshop on Software Specification and Design* (Como, Italy), pp. 140-147, Oct. 1991.
- [18] Kawata, K., Takura, A., Ohta, T., "On a Communication Software Generation Method from Communication Service Specifications Described by a Declarative Language," in *Proc. Fifth International Conference on Computing and Information* (Sudbury, Canada), pp. 116-122, May 1993.
- [19] Liu, M. T., "Protocol Engineering," *Advances in Computers*, vol. 29, Academic Press, 1989.
- [20] Bochmann, G. v., "A general transition model for protocols and communications services," *IEEE Trans. Comm.*, COM-28(4), pp. 643 - 650, 1980.