

〔公 開〕

TR-C-0116

通信サービス仕様 S T R から
プログラム仕様 S D L への変換

深山 尚
Hisashi MIYAMA

1 9 9 5 4 . 1 4

A T R 通信システム研究所

通信サービス仕様 STR からプログラム仕様 SDL へ
の変換

深山 尚

1995 年 4 月 14 日

目次

1	はじめに	3
2	通信ソフトウェア設計手法	3
3	通信システム仕様記述言語 STR	3
3.1	グラフ文法	4
3.2	規則の優先順位	4
4	通信プログラム仕様記述言語 SDL	5
4.1	自動生成する SDL の特徴	5
4.1.1	プロセス間通信による広域状態遷移の実現	6
5	STR 仕様, SDL 仕様の関係	8
5.1	SDL 仕様から STR 仕様への変換方法	9
6	自動生成のアルゴリズム	9
6.1	通信経路の決定	10
6.2	GP から LP への変換, 単位遷移動作の生成	11
6.2.1	規則グラフ中の点近傍	11
6.2.2	単位遷移動作の生成	11
6.3	LP から LC への変換, 単位遷移動作の合成	12
7	おわりに	13
7.1	結果	13
7.2	今後の課題	14

1 はじめに

通信サービスに対する仕様記述言語として、SDL[1]やLOTOS [2]など数多く存在する。SDLでは、サービスを実現するプロトコルを規定することにより、通信サービスを規定している。LOTOSでは、お互いに相互作用を行うサブプロセスの集合からなるプロセスとしての対象をモデル化している。プロセスはブラックボックスとして扱っているが、プロセス間の同期を仕様として記述するので、通信システムの外部から認識可能な仕様だけでは完全な仕様を記述することはできない。

ここで提案するSTR [3, 4, 5]は以下の特徴を持っている。外部から認識可能な端末の状態変化を仕様として記述することにより、通信システムの詳細知識を持たない通信システムの非専門家でも通信サービス仕様を設計することができる。

STRによって書かれた通信サービスを実機上にて実現するため、プログラム仕様へと変換を行わなくてはならない。状態遷移図を表示でき、一般に使用されているSDL仕様をプログラム仕様とする。本稿では、STR仕様からプログラム仕様であるSDL仕様へ自動変換する手法を提案する。プログラム仕様へと変換された通信サービスは、プログラミング言語へと変換される。この自動変換により、通信ソフトウェア開発の短縮化、簡略化を図ることができる。

次の章では、これらの通信ソフトウェア設計手法について述べ、3章では、外部から認識可能なサービス記述言語について述べる。4章では、通信プログラム仕様記述言語を述べ、5章では、サービス記述言語、通信プログラム記述言語の関係について述べる。6章より具体的にサービス記述言語から通信プログラム仕様への変換について述べる。7章にて自動変換のまとめと今後の課題について述べる。

2 通信ソフトウェア設計手法

STR-SDL自動変換システム [6, 7, 8]では、通信サービス記述であるSTR仕様からプログラム仕様への自動変換が行なわれている。図1で示すように、非専門家対象のSTR仕様を使用し、グラフィカルなツールにて対話しながら通信サービスの内容を具体化してゆく。書かれたSTR仕様に矛盾や抜け落ちがないように各種検証ツール、補完ツールもサポートしている。このように要求された通信サービスを満足するSTR仕様から、システムを構成する個々のエンティティの振る舞いを表現するプログラム仕様への自動変換がなされる。変換されたプログラム仕様は文書としての価値をもち、さらに実機を稼働するためにC言語であるプログラムへと変換される。

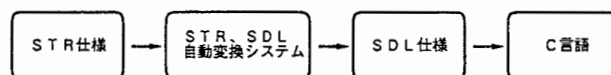


図 1: 通信ソフトウェア開発環境

3 通信システム仕様記述言語 STR

STR[3, 4, 5]では広域状態を論理式で表現し、広域状態遷移を、プロダクション規則 (STR規則)の集合で記述する。STR規則は、始広域状態、イベント (端末入力)、次広域状態の3項で表現される。広域状態は複数の端末の状態から構成される。端末の状態及びイベントは述語で表現される。

通信サービスの動作をSTR記述で記述したものを図2に示す。この規則は、ダイヤルトーン状態にある端末Aから、空 (idle) 状態である端末Bにダイヤルしたとき、端末Bが着信転送を設定 (m-cfv(B,C)) しており、その転送先である端末Cが空状態の時には、Bへの呼びは、Cに転送され、Aは呼返音 (ringback-tone) 受信状態へ遷移し、Cは呼出音 (ringing-tone) の鳴っている状態に遷移することを規定している。

図2の規則で、dialtone, idle等は端末の状態を表現する述語であり、dialはイベントを表現する述語である。なお、一つの端末が同一の述語を2個以上持つことはないものとする。

```

dialtone(A), idle(B), m-cfv(B,C), idle(C) /* 始広域状態 */
dial(A,B):                               /* イベント */
ringback(A,C), ringing(C,A),
pingring(B,A), m-cfv(B,C).               /* 次広域状態 */

```

図2: 通信サービス規則例

3.1 グラフ文法

STR規則中の始広域状態記述とイベント記述は、あわせて1個のラベル付きの有向グラフとして表現できる。また、次広域状態記述も1個のラベルつき有向グラフとして表現できる。

図2の規則に対応するグラフ表現を図3に示す。図3において、規則中のringback(A,C)という記述は、点Aから点Cへのringbackというラベルを持つ有向辺として表現される。

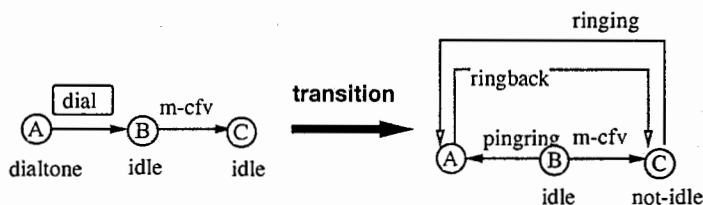


図3: サービス規則に対するグラフ表現

このように、STR規則は、グラフの書き換え規則とみなすことができる。通信システム全体の状態も、グラフで表現できる。これをシステム・グラフと呼ぶことにすると、STR規則の意味するところは、次のように定義できる。

STR規則の意味

図4にあるようにSTR規則は、システム・グラフ中の部分グラフで規則の始グラフと同形なものを、規則の次グラフと同形なものに置き換えることを指定する。

3.2 規則の優先順位

イベントが生起したとき、複数の規則が適用可能になる場合がある。このような規則間の競合を回避するため、規則間に優先順位をつける。ここでは、同一イベントを持つ規則の集合に対して、グラフの包含関係に基づいた次のような半順序を導入する。同一のイベントを持つ2つの規則r1,r2に対して、r1の始グラフがr2の始グラフを含む時その時に限り、規則r1は規則r2よりも優先度が高い。この順序は全順序ではないため、規則の競合を完全に回避することはできない。上の半順序を用いても規則の競合を回避できない時、適用可能な規則のどれか1つが実行されるものとする。

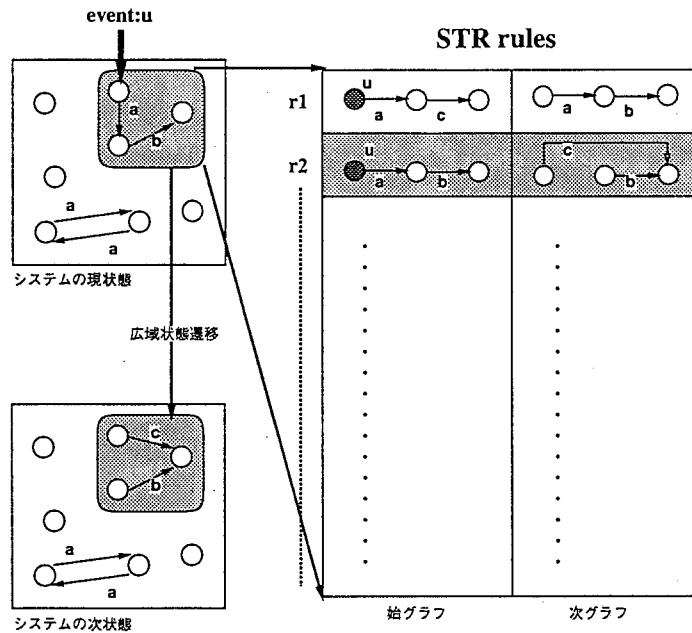


図 4: STR 規則の適用

4 通信プログラム仕様記述言語 SDL

SDL[1] は、通信システムの機能に関する仕様や動作を明確に記述するため、国際電信電話諮問委員会 ITU-U にて開発し勧告として標準化したものである。おもに交換機や通信端末などの機能仕様の記述に使用されている。

SDL にはテキスト表現方式 (PR) とグラフィカル表現方式 (GR) の 2 つが存在する。ここでは、グラフィカル表現方式について説明する。図 5 の上部で GR の 4 つの構成要素について示している。State とは状態のことで、例として、idle 状態、path 状態などを示す。Input は、イベント入力や、信号入力を示す。Output では信号出力を示す。最後に DT (Decision Table) は SDL では定義されていない。しかし、自動変換する SDL 中に便宜上使用している。DT は条件により分岐を行うものである。自動変換のシステム内では「同型判定部」と呼ぶこともある。

設計者が SDL を使用し意図したプログラム仕様を製作する。このプログラム仕様を SDL 仕様動作層のプロセス上にて動作させる。SDL 仕様動作層上では、通信処理を実行する単位をプロセスと呼び、一般に記述対象のシステムを複数のプロセスから構成する。各プロセスは拡張有限状態遷移機械としてモデル化され、他プロセスとの間で信号を送受することが可能で、信号を入力することにより、その入力情報と基の状態に応じて必要な処理を実行し、次の状態へと遷移する。この信号送受において、各プロセスは非同期的に動作する。

4.1 自動生成する SDL の特徴

自動変換システムにより変換される SDL は以下の特徴をもっている。

プロセスの動き

SDL 仕様上のプロセスと STR 仕様上の端末が 1 対 1 であり、それぞれのプロセスは同形の通信プログラム仕様動作している。プロセスは初期状態から始まり初期状態へ戻る必要がある。図 6 に示してあるように、STR 言語の記述は外部認識可能な端末の状態変化を引き起こす入力に対してイベントを割り付け、端末の状態変化として仕様を記述して

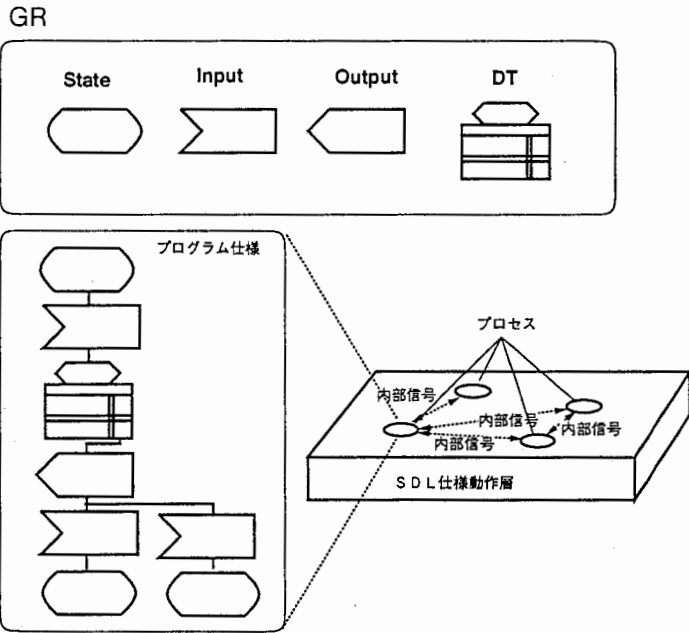


図 5: 通信プログラム仕様 SDL

いる。STR から SDL へ変換し変換される SDL の動作形態は、STR 仕様の端末上の振る舞いをもとに生成している。そのため、SDL 仕様動作層上のプロセスの動きは STR 仕様の端末の振る舞いと同じでなければならない。

4.1.1 プロセス間通信による広域状態遷移の実現

各々のプロセスは局所的な情報しか保持しない。プロセスは、通信することによって周辺プロセスと情報を交換し、広域的な状態を把握して、適用できる STR 規則を決定する。

プロセスの状態

各時点におけるシステム中のプロセスの状態は、変数を含まない素式の連言で記述できる。各素式の第 1 項には、そのプロセス自身の識別子 (PID) が入る。たとえば、識別子 p_1 を持つプロセスで、ダイアルトーンが鳴っていて、識別子 p_2 を持つプロセスへ着信転送が設定されている状態は、次式で記述できる。

$$\text{dialtone}(p_1), \text{m-cfv}(p_1, p_2)$$

このように状態記述中に PID を含むので、プロセスの状態数はシステム中に存在するプロセスの個数に依存する。しかし、状態記述中のプロセス ID を無視すれば、プロセスの状態はシステムに依存せずに STR 記述だけから、有限個に分類できる。その分類番号を状態 ID と呼ぶことにすると、プロセスの状態は、状態 ID と PID の列で表現できる。

内部信号の内容

プロセス間で送受信する内部信号として、探索依頼信号とそれに対する応答信号の 2 種類用意する。

探索依頼信号は、自分の状態を隣接プロセスに伝えることによって、隣接プロセスに適用できる規則の決定を依頼するために用いられる内部信号であり、応答信号は、決定された規則を通知するために用いられる内部信号である。応答信号の特殊なものとして、拒否信号がある。これは、適用できる規則が存在しないことを意味する内部信号である。

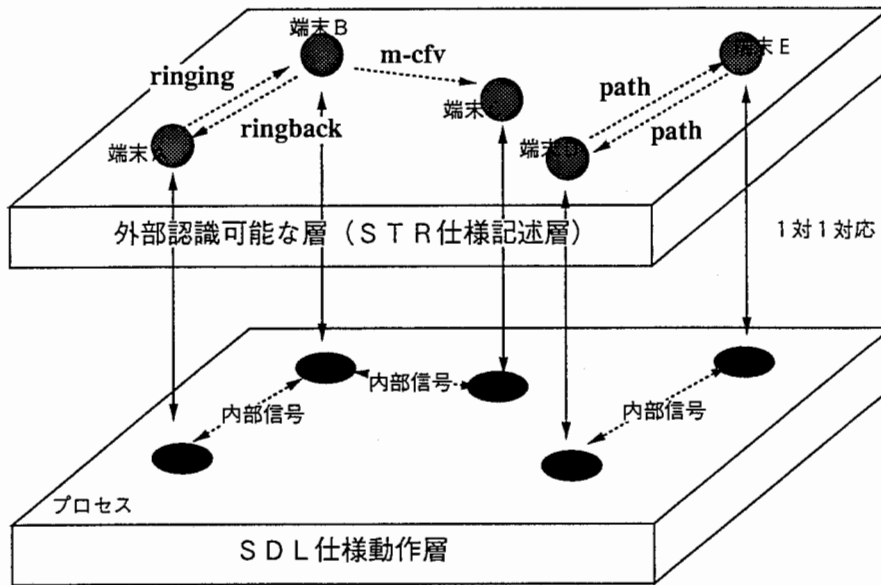


図 6: SDL 上のプロセスと端末の動き

探索依頼信号は、適用できる可能性のある規則の集合を陰に表している。順次、探索依頼信号が伝えられていくに従って、適用できる可能性のある規則が絞り込まれ、最後に規則が一意に決定される。

内部信号はプロセスの状態を伝達する。そのため、内部信号中には内部信号 ID とは別に内部信号を発信したプロセスの保持するプロセスの ID が含まれる。

プロセス間通信

信号には、探索依頼信号と応答信号がある。探索依頼信号には、それまでの探索で決定した規則や、調査すべき適用候補規則、探索済みプロセスのプロセス ID 等の情報が積まれている。一方、応答信号には、各プロセスが状態遷移を行うための情報等が積まれている。各プロセスの状態遷移は、探索依頼信号と応答信号の授受で決定される。

上記の信号を用いたプロセス間通信の例を図 7 に示す。プロセス間通信での問い合わせの順序は、規則グラフの重ね合わせによって得られた仮通信経路によって決定される。仮通信経路の順に探索依頼信号が送信され、各プロセスでの比較/演算処理、あるいは情報の再設定等によって、適用可能な規則のしぼり込み決定がなされる。

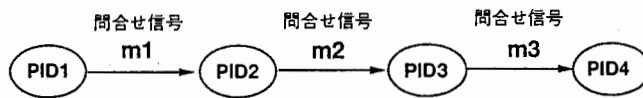
図 7 に示す仮通信経路の順に探索依頼信号が送信され、最終プロセスが各プロセスに回答信号 (respond: 規則 ID) を送信している。今決定した規則を r とするとプロセス PID 1, PID 2, PID 4 が規則 r によって状態遷移しプロセス PID 2 が状態遷移しないとき、最終プロセス PID 4 は PID 1 と PID 3 に対して遷移を行う情報を載せた回答信号 (respond: r) を送信し、さらに遷移に関係しないプロセス PID 2 に対して、状態遷移を行わない旨の情報を載せた拒否信号 (respond:norule) を送信している。

プロセス ID の送信

図 8 は、図 7 中のプロセス PID 3 が状態 s_1 で PID 4 から回答信号 respond: r を受信した後、respond: r の指定する規則 r_1 に従って状態 s_2 へ遷移する状況を示している。

m_2 中には PID1, PID2 の保持する PID が含まれ、respond: r 中には遷移すべき状態をあらわしている遷移指示命令が含まれる。 s_1 から s_2 に遷移する際に、保持している PID を m_1 と m_3 中の PID 情報を利用して再設定する。

仮通信経路



プロセス間通信

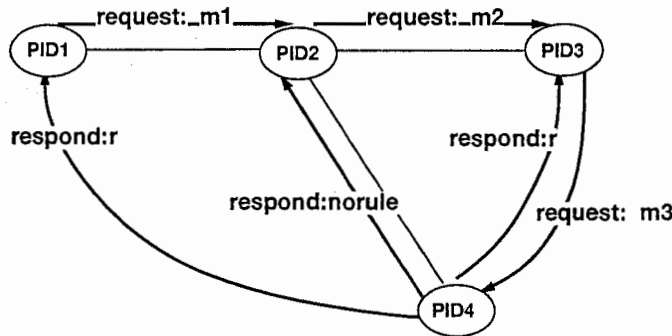


図 7: プロセス間通信例

5 STR 仕様, SDL 仕様の関係

STR 仕様と SDL 仕様の関係および特徴, SDL 仕様から STR 仕様への変換過程を明記する。

部分適用はプロダクションシステムでは一般的なルール適用方法であり, STR 仕様も同様な特徴を持っている。適用条件部を満足している場合, すなわち, ネットワーク内の状態 (実状態と呼ぶ) が STR 規則の適用条件部を包含している時に STR 規則が適用され状態遷移がおこる。これに対し, SDL 仕様では実状態と SDL 仕様で記述されている状態が完全一致しているときのみ状態遷移が行われる。このような規則適用方法を完全適用と呼ぶ。完全適用では, 遷移する状態を全て把握しなければならないのに対し, 部分適用では必要な条件のみを把握すればよいというメリットがある。

また, ネットワーク内複数端末の状態遷移を記述する方法をグローバル記述, ネットワーク内単数プロセスの部分的な状態遷移を記述する方法をローカル記述と呼ぶ。グローバル記述で通信仕様を記述する場合, STR 仕様にて通信サービスを記述する場合, 通信プロトコルに関する知識が不要である。それに対し, ローカル記述である自動生成する SDL 仕様にて通信サービスを記述する場合, 通信プロトコルに関する知識, 具体的にはネットワーク全体の状態を知るために, 各々の状態を1つにまとめるために信号のやり取りをする。このやり取りをプロセス間通信で行っている。

したがって, ネットワークの非専門家にとって, ネットワーク全体を記述するにはグローバル記述で部分適用の言語仕様がふさわしいと考える。

仕様記述法を適用法の観点から部分一致適用と完全一致適用に, 記述範囲の観点からグローバル記述とローカル記述に分類した。それぞれの特徴を組み合わせて次の4種類の記述法が考えられる。図9にて説明する。

- GP (Global Partial)

グローバル記述でかつ部分適用とする言語仕様。今回の変換過程では STR 仕様を示す。

- LC (Local Complete)

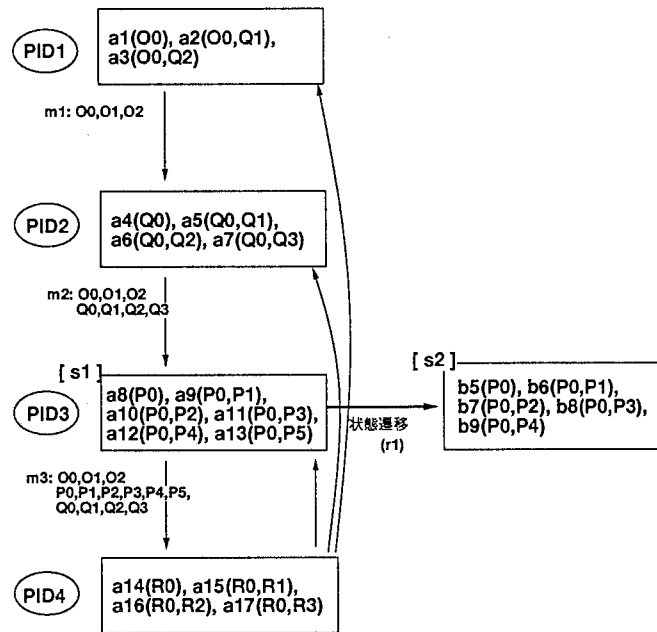


図 8: PID の送信例

ローカルな記述でかつ完全適用とする言語仕様。今回の変換過程では SDL 仕様を示す。

- LP (Local Partial)
ローカルな記述でかつ部分適用とする言語仕様。今回の変換過程では単位遷移動作と呼ぶ。
- GC (Global Complete)
グローバルな記述でかつ完全適用とする言語仕様。今回の変換過程では存在しない。

5.1 SDL 仕様から STR 仕様への変換方法

図 9において、以下の経路にて変換を行う。

$$GP(\text{STR 仕様}) \Rightarrow LP \Rightarrow LC(\text{SDL 仕様})$$

LP 経由では、部分適用化を LP から LC への最後の段階で行うので、管理する状態数が少なくてすむ。したがって、LP 経由の方法を採用する。

6 自動生成のアルゴリズム

SDL 仕様の変換過程は次の 3つのステップに分けることができる。

1. 通信経路の決定

STR 規則上のイベント生起点から、そのイベントに関する STR 規則上の端末を全て経由する通信経路を見つける。

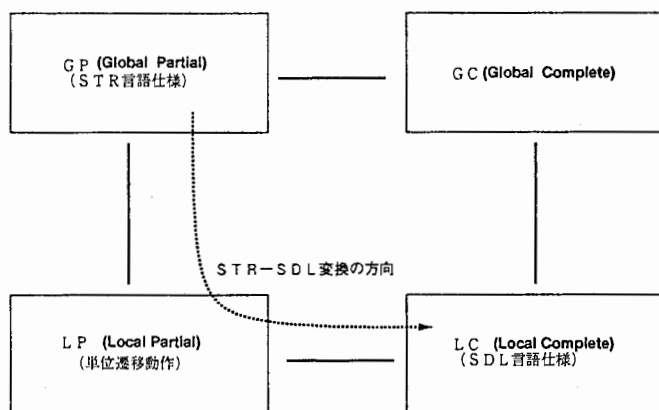


図 9: STR, SDL 変換概要

2. 単位遷移動作の生成. GP から LP への変換

グローバル記述されている STR 仕様より, ローカル記述している SDL 仕様への変換をおこなう. それには, 以下の3つの手順を要する.

(a) 規則グラフ中の点近傍

STR 仕様の規則上の各端末の相互関係をローカル記述化する. 具体的には, STR 仕様の各々の規則をグラフで表現する. これを規則グラフと呼ぶ. さらにそれぞれの端末ごとの規則グラフを作成し近傍を整理する. 複数プロセスが関係している相互関係を明確にするためである.

(b) 単位遷移動作の生成

各々の端末ごとの近傍には3つのことがらが存在する.

- ローカルな現状態.
- STR 仕様で意図した規則形態と, 各々端末が SDL 仕様上で同じ形態を取っているかの判別作業をおこなう.
- もし他の端末へプロセス間通信を行うならその方向をしめす.

3. 単位遷移動作の合成. LP から LC への変換

ここで, 実状態を生成する. 生成できた単位遷移動作を合成し, 初期状態から始まり, 初期状態へ戻る SDL 仕様を作る.

6.1 通信経路の決定

イベント生起点より, 同一イベントで通信システム上の状態と STR 規則の現状態と同一な STR 規則を見つけるのに, 最適な通信経路を見つける.

与えられる STR 規則を有向グラフ表現する. 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生起とするグラフの書換え規則とみなせる. 従来, 適用条件を表すグラフに分岐が無いという制限 [10, 11] が設けられていたが, 本稿では仮想的に分岐のないグラフに変換することにより, 最適な通信経路 (仮通信経路と呼ぶ) を見つける.

仮通信経路は STR 規則の同一イベントで, かつ包含関係を用いて作成した重ねグラフに, ラベルの包含関係を導入して作成した重ねグラフ上に出現する端末の出現順に, 規則グラフ上の各頂点を並べたものである. この仮通信経路上で表現する端末は, 状態記述要素の第一引数になっているもののみとする. 仮通信経路上に現れる端末ごとのプリミティブの増減情報, 端末の隣接関係や信号テーブルの情報を付加して, 仮通信テーブル上で関数の形として出力する [8, 9]. この仮通信テーブルとは, 実際の通信システム上で動作するための制御情報である.

6.2 GP から LP への変換，単位遷移動作の生成

GP から LP への変換を行う。変換の主な目的はグローバルで記述してある STR 仕様をローカルな記述に変換することである。ここでローカル記述への変換とは，単位遷移動作への変換のことで，各々の状態はローカルな状態であり，広域状態を把握するために内部信号を持つものである。

6.2.1 規則グラフ中の点近傍

STR 規則の始グラフにおいて，ラベルを持つ点はイベント生起点を先頭にして一列に並んでいる（仮通信経路）。この並びに応じて幹中の点の間に関係を設定できる。この関係に基づいて，規則グラフ中の点の近傍を以下の4種類導入する。図 10， 11。

現状態近傍 v と v から出ている辺及びそれらに付随するラベルで構成される近傍である。単純近傍はプロセスの状態を表現している。イベントも単純近傍に含まれる。

同型判定近傍 v の単純近傍の各辺に対して，その辺の終点 w が v の祖先を始点とする辺の終点になっている場合，祖先の種類（親，祖父等）と祖先を始点とする辺に付随しているラベルを単純近傍に付加したもの。 w が複数の祖先に対して，それらを始点とする辺の終点となっている場合，もっとも関係の近いプロセスから出ている辺を優先し，その情報だけを付加する。具体的には，図 12 上の違いを判定するものである。

通信経路近傍 仮通信経路上に示してある，次に探索するプロセスを表す。

次状態近傍 次状態での v と v から出ている辺及びそれらに付随するラベルで構成される近傍である。

6.2.2 単位遷移動作の生成

単位遷移動作はある1つのプロセスが，ある状態（現状態）から次の状態（次状態）に遷移する間に，採りえる状態の流れである。図 13。

現状態 あるプロセスの局所状態を意味する。ここで局所状態とは，該当するプロセスを第一引数とする状態記述子の集合で，上記の単純近傍のことである。

信号受信 外部イベント入力または，別プロセスからの探索依頼信号を受信する。この信号には該当するプロセスが次に採りうる状態遷移情報（候補規則集合）等が含まれている。

同型判定部 信号を受信したら，プロセス ID の比較を行う。上記の同型判定近傍のことである。

信号送信 1 他プロセスに関して調査を行わないと，遷移すべき形態が決定できない場合，仮通信経路で決定された次の探索先プロセスに対し探索依頼信号を送信する。

信号送信 2 該当プロセスで探索が終了したか，あるいはその時点で確定している適用可能な遷移が適用可能遷移情報優先順位が高い物であった場合，探索を終了し，その遷移に該当するプロセスに対し，応答信号を送信する。このとき遷移に関わるプロセスに対しては，遷移すべき状態を示す応答信号を送信し，遷移に関わらないプロセスに対しては，遷移拒否の命令を表す拒否信号を送信する。

応答受信と拒否受信 他のプロセスから，応答信号を受信する。この時応答信号の内容が，遷移を行うよう指示するものであった場合，確定した遷移規則に基づき，次状態に遷移する。あるいは，遷移を行わない旨の信号を受け取った場合は，処理を中止して現状態に戻る。

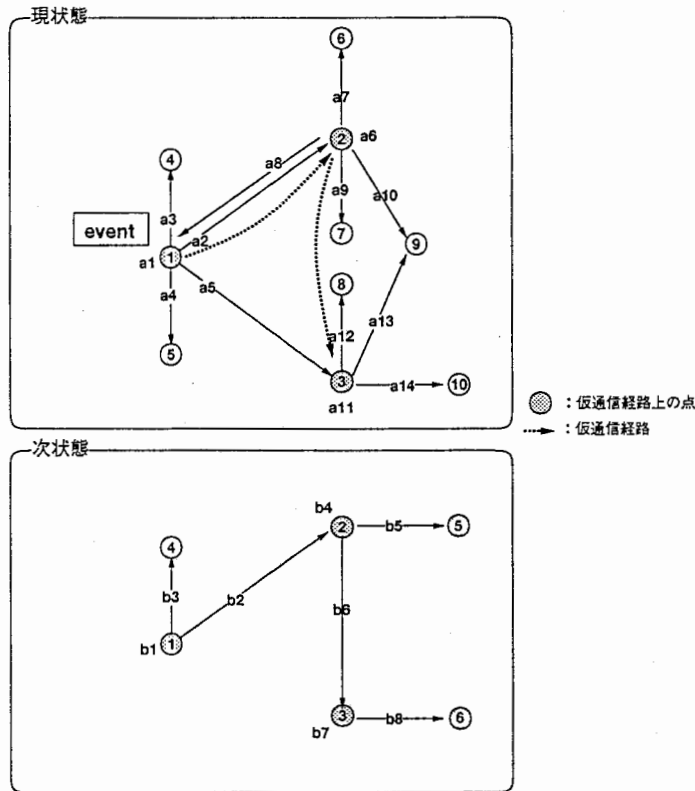


図 10: STR 規則のグラフ表現

PID 再設定 現状態と応答信号の内容から、該当プロセスの次状態で用いるプロセス ID を設定しなおす。

次状態 あるプロセスが遷移すべき局所状態を意味する。上記の次状態近傍のことである。

6.3 LP から LC への変換，単位遷移動作の合成

生成された単位遷移動作を用いて、初期状態から到達可能なすべての状態を生成し、各状態におけるプロセス動作を生成する。図 14。

(1) 初期状態の展開。

ステップ 2 で生成された単位遷移動作の内、初期状態を現状態として持つパターンを集め、初期状態時の動作を合成する。

(2) 次状態の展開。

(1) で生成された遷移動作の全ての次状態 s に対して、ステップ 2 で生成された単位遷移動作の内、 s に含まれる状態を現状態として持つものを集め、そのパターンと (1) で生成された単位遷移動作とを合成する。

図 14 に次状態の展開を例示する。図中のパターン 1 は、アイドル状態から展開が進んで、状態 s_1 を現状態とするパターンが張りついているところである。図中のパターン 2 は、現状態 s_3 が s_1 に含まれ、パターン 1 と同じ内部信号 m_1 を受信した場合の単位遷移動作である。パターン 1 中には、応答信号の一部である拒否信号を受信し

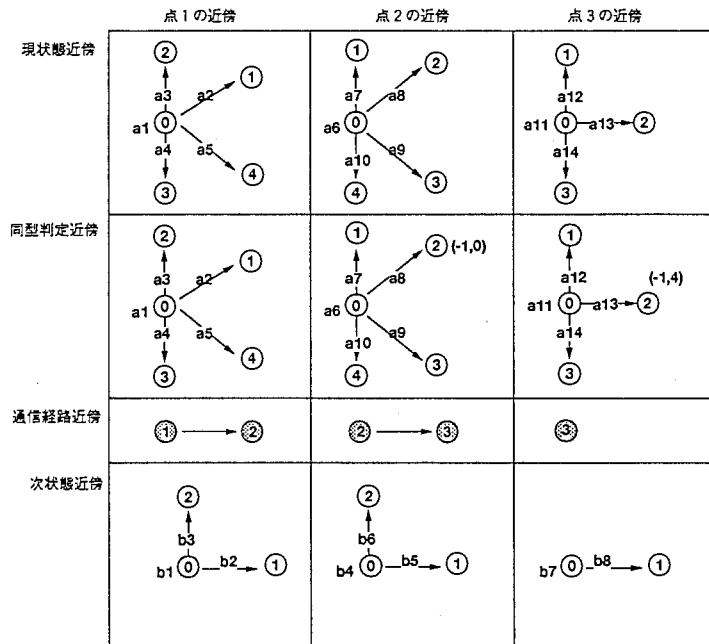


図 11: グラフ中の点近傍

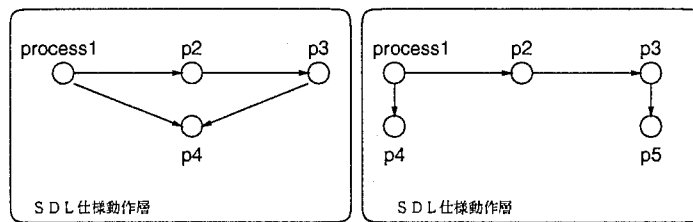


図 12: 同型判定部の意義

て現状態に戻る動作が必ず含まれている。パターン1中の拒否内部信号受信後の動作を、パターン2中のm1受信後の動作に置き換える。この時付加されるパターン2の次状態s4をs5に書き換える。s5はs4にs1とs3の差分(s1-s3)を付加した状態である。こうして、新しい状態が生成される。

合成されたパターン3を(1)にもどして、(2)の次状態を展開を新しい状態が生成されなくなるまで繰り返す。

7 おわりに

7.1 結果

グラフの書き換え規則として表現される通信サービス仕様から、これを実現するプロトコルを生成するアルゴリズムを示した。これにより、通信システムの内部を知らない非専門家であっても通信プログラム仕様を生成することが可能となる。

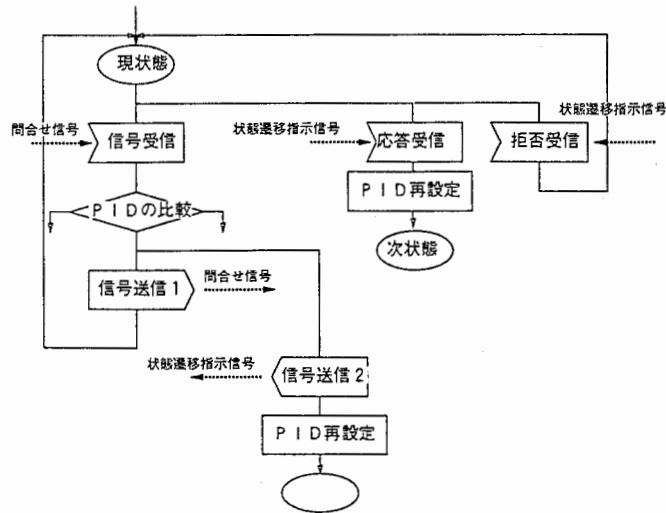


図 13: 単位遷移動作

一般に部分グラフ同型判定問題は NP 完全問題であることが知られている。このため、大きな規則グラフに対して同型な部分グラフを効率的に探索することは、ほとんど不可能である。

しかしながら、実用上は問題が無いと考える。すなわち、電話サービスをはじめとする通常の通信サービスでは、1つの呼にかかわる端末の数はそれほど多くなく、組合せ的な爆発はほとんど起こらない、そのような理由によるものである。

7.2 今後の課題

SDL で書かれた通信サービスを STR へ逆変換することが考えられる。STR 仕様上の通信サービスが SDL に変換され、テストする。この時、SDL 仕様上でのサービスの洩れや、間違いが発見されることもある。SDL 上にて修正を行うのはもちろんだが、同時にサービス仕様である STR 規則も修正しなければならない。これでは、二度手間となる。そこで、修正した SDL 仕様を STR 仕様へ逆変換するシステムを考える [12]。これにより、プログラム仕様への変更をサービス仕様に自動的に反映する手法を確立できる。

この逆変換システムは、図 15 に示すように部分適用化とグローバル記述化の 2 つの処理から成り立っている。

部分適用化 完全適用である SDL から STR の特徴である部分適用へ変換を行う段階的手法を確立した。

共通部分抽出法 これは複数遷移より共通のイベント、状態を導き出す方法である。

事例ベース手法 さらに、共通部が存在しない場合も提案する。これは、事例ベース上より部分適用がなされている部分を導き出す方法である。

これにより正確な部分適用化がなされ、ローカルな記述形態であるが STR 仕様と同様な部分適用記述がなされる。

グローバル記述化 現在自動生成している SDL は外部認識可能な端末に対応した各プロセス上で動作する目的で生成されている。そのため各プロセスの状態と接続関係をもつプロセスの関係しか状態上に記述されていない。このような記述をローカル記述と呼ぶ。しかし、STR では、複数の端末の関係が記述されている。これをグローバル記述と

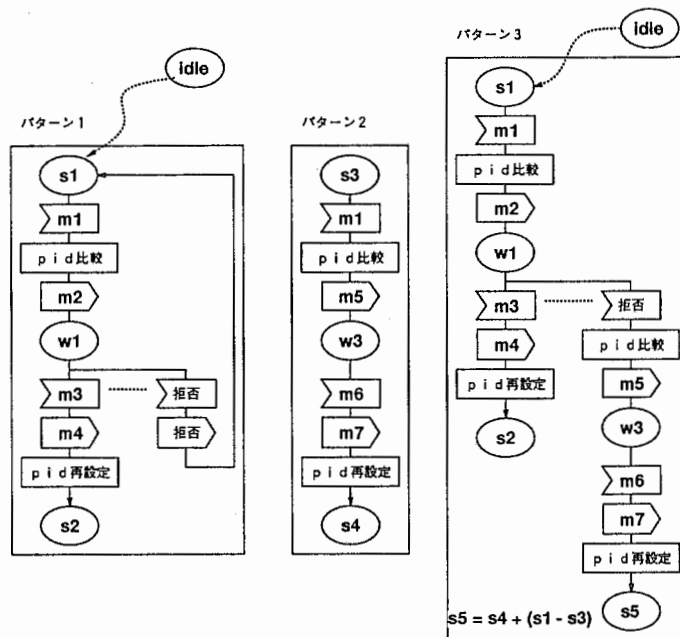


図 14: 単位遷移動作の合成

呼ぶ。部分適用化処理で出力した部分適用記述から、グローバルな記述仕様へ変換をおこなう。

内部信号の動的解析 イベント生起点のプロセスから各々のプロセスへの内部信号を解析し、グローバル記述上でのプロセスの ID を決定する方法である。

この逆変換の問題点は、既存サービスの STR 規則と逆変換した STR 規則が同一とは限らない点である。

既存の STR 規則と逆変換した STR 規則は修正したサービス部分が違うのは当然だが、既存通信サービスの内容は同一である。しかし、表現上記述の違いを生じる可能性がある。これは、STR 規則の再利用時に同一通信サービスであることを保証しなければならず、STR 仕様での通信サービスの管理ができなくなる可能性がある。STR 仕様での通信サービスの

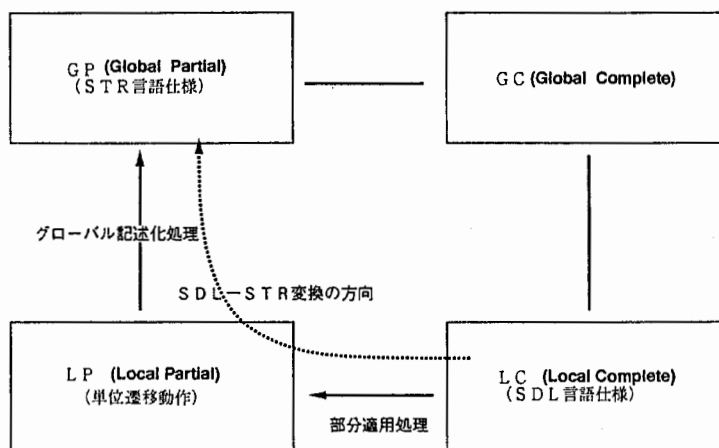


図 15: 逆変換の過程

管理とは、既存の自動変換のシステムでは、STR仕様とSDL仕様の2つで通信サービスを管理していたが、SDL仕様への変更をSTR仕様へ自動的に反映できる手法が確立できることにより、下流工程中の手戻りをSTR仕様へ自動的に反映することができるようになることである。しかし、下流工程中の手戻りにより、既存のサービス記述との記述上の食い違いが生じると、同一サービスにもかかわらず他サービスと認識してしまう可能性がある。そこで記述上も同一サービスなら同じにする必要がある。

参考文献

- [1] ITU-T: Recommendation Z.100, "Functional Specification and Description Language (SDL)," 1992.
- [2] T. Bolognesi, E. Brinksma, "Introduction to the ISO Specification Language LOTOS," Computer Networks and ISDN Systems, 14, pp. 25-59, 1987.
- [3] Y. Hirakawa, T. Takenaka, "Telecommunication service description using state transition rules," Proc. Sixth Int. Work. Software Specification and Design, pp. 140-147, Oct. 1991.
- [4] Y. Hirakawa, K. Kawata, T. Takenaka, "Rule descriptions for telecommunication services and their transformation into standardized specification descriptions," IEEE 8th International Conference on Software Engineering for Telecommunication Systems and Service, Apr. 1992.
- [5] 平川 "STR (State Transition Rule) 記述仕様書" ATR テクニカルレポート, TR-C-0073, 平成4年1月
- [6] 深山尚, 田倉昭, 太田理「通信サービス記述からプログラム仕様への自動変換と最適化」電子情報通信学会 交換システム研究会 SSE93-101, 1993年11月.
- [7] 河田慶三, 田倉昭, 太田理「端末の動作記述からプロセス仕様を自動生成する手法について」電子情報通信学会 交換システム研究会 SS92-21, 1993年1月.
- [8] 田倉昭, 太田理「グラフ表現可能な通信サービス仕様からソフトウェア仕様への変換法について」電子情報通信学会全体, 1994年9月.
- [9] 田倉昭, 世良孝文, 太田理「グラフ変換を用いたプロトコル合成法」電子情報通信学会 コンピュータサイエンス研究会 COMP94-102, SS94-64, 1995年3月.
- [10] 河田, 平川, 竹中 "分散システムの記述とプロセス動作" 情報処理学会全国大会, 4Q-5, 平成2年9月
- [11] 河田, 平川, 竹中 "通信サービス記述からプロセス動作仕様への変換の概要", 電子情報通信学会全国大会, B-535, 平成3年3月
- [12] 深山尚, 田倉昭, 太田理 "通信プログラム仕様からサービス仕様への逆変換手法," 電子情報通信学会交換研究会, SSE94-257, 平成7年3月