5T-7

マッチング機能を利用した通信システム仕様記述手法

平川　豊
NTT　電気通信研究所

1．はじめに

通信サービスの仕様記述では、有限状態遷移図を基本とし、状態にメモリを持たせるなどして記述能力を強化した手続的記述手法(SDL,Estelle など)が利用されている。しかし、多者の関わるサービス、特に不特定多者の関わるサービスに関しては、従来手法による簡潔な記述はできない。

本稿では、マッチング(変数を含むリストの整合性判定)機能を利用した非手続形[1,2]の仕様記述手法を提案する。この手法により、不特定多者の関わるサービスの簡潔な記述が可能である。本手法では、マッチング機能を用いることにより、複数の状態遷移規則を1つのパタン遷移規則で記述可能とし、無限個の状態遷移規則をも記述可能として、記述能力を強化している。

以下では、不特定多者の関わることのできるキャッチホンサービス記述を例に取り上げ、提案する記述手法による記述例を示す。次に従来手法による記述との比較・検討を行う。

2．不特定多者の関わるキャッチホンサービス

発端末をAとし、着端末をBとする。AとBの通話中あるいはAがbusy tone状態のとき、端末C，D，E,..が端末Aに対して次々に発呼するものとする。このとき、端末Aからの以下の信号により、不特定多数の人とかわるがわる通話できるサービスをキャッチホンサービスと定義する。

onhook: 端末Aを idle 状態とし、サービスに関わっていた全端末を busy tone状態とする信号である

reject: 端末Aに新たな端末からの着信要求があったとき、サービスに加わることを拒否する信号である

accept: 端末Aに新たな端末からの着信要求があったとき、サービスに加わることを許可し、現在通話中の相手を待ちキューにつなげ新たな端末との通話を行なう信号である

roll: 通話の相手をキューの先頭で待っている相手と交換する信号である

他端末からの信号は、例えば、onhook(id)と書き表わす。ここでidは信号の発端末を特定する数値である。

サービス実現のためのプログラムは、各端末毎に独自のプロセスを持つものとする。端末Bと通話中の端末Aは、端末Cからの着信要求を、端末Cの制御プロセスからの内部信号 income(C)の受信によって認識するものとする。

以下では、上記のキャッチホンサービスにおける、端末Aのプロセスのうち、3者以上に関わる部分を記述対象とする。

3．マッチング機能を利用した仕様記述手法

(1)記述法の説明

本手法では、状態を文字定数(小文字)と変数(大文字)を要素とするリストで表現し、任意個の状態をマッチング(リストの整合性判定)機能を用いて1つのパタンで代表させる。例えば[1 2 2] [1 3 3]等で表現される状態は、パタン[X Y Y]で代表する。これは2番目と3番目が同じ要素である3つの要素からなるリストを表現している。あるいは、これらの状態を1を先頭とするリストという意味でパタン[1..]で代表してもよい。

本記述法では、このパタンによる表現を用い、任意個の状態遷移規則を1つのパタン遷移規則として記述する。パタン遷移規則はパタン遷移関数δを用いて以下のように表現できる。

$$\delta(P_i, sig(5)) = P_{i+1}$$

この遷移規則は、パタンP_iにマッチ(リスト表現として整合)する状態で信号sig(5)を受信したとき、パタンP_{i+1}で示される状態に遷移することを表わす。

簡単な例として、3つの sig(i) 信号(iは何でもよいが、3つの信号について同じでなくてはならない)により動作が進行するシステムの記述例を示す。

δ(start,sig(X)) = [X X]
δ([Z..] ,sig(Z)) = [..]
δ(nil,λ) = ×

startは初期状態を示す。2番目の規則は、1以上の長さのリストで表現される状態から、その先頭要素を除いたリストで表現される状態への遷移を意味する。またλは空入力を意味し、3番目の規則は、パタンがnilとなった時点で動作が終了することを表現する。

具体的動作は次のようになる。例えば、初期状態で sig(3) という信号(信号が伴う数値3は変数 Xに代入される)を受信した場合には、[3 3] で表現される状態に遷移する。[3 3] の状態は、Z = 3 を先頭の要素とするリスト表現とみなすことができるので、再び信号sig(3)を受信した場合には2番目の規則の適用が可能であり、適用されると[3] で表現される状態に遷移する。その後、この規則がもう1回適用された後に、3番目の規則が適用され、このシステム動作は終了する。

パタン遷移規則の記述によって、実際には無限個の状態遷移規則が記述可能であるため、本記述手法の適用は有限状態機械に限定されたものではない。

(2)キャッチホンサービスの記述例

記述例を図1に示す。図1で、[talk X]とは Xで示される端末と

Description Method using Matching Function for Communication Service Specification

Yutaka HIRAKAWA

NTT Electrical Communications Laboratories

通話中であることを、また[intr Y talk X] とは Xとの通話中に Y から着信要求があることを表現している。また、[wait Z]は、端末 Z のユーザが待ち状態であることを意味している。

マッチング機能を利用することにより、不特定多者の参加できるキャッチホンサービスは、遷移枝数２６で記述できた。

４．従来記法との比較

ここでは、通信用仕様記述言語として従来から知られている SDL (Functional Specification and Description Language)[3]による記述との比較を行う。SDL は状態遷移図を基本としているが、任意の手続記述が可能なフローチャート記述言語である。キャッチホンサービスの記述イメージを図２に示す。遷移枝数は４８である。

待ち合わせ中の端末から onhook 信号を受信した場合の動作に関し、図１と図２の対応する部分を斜線で示した。図２では、①通話中の状態(talk_withQ)、②busy tone の状態(busy_withQ)、③通話中で新たな端末からの着信要求がある状態(talk_withQ_intr) 、④ busy tone の状態で新たな端末からの着信要求がある状態(busy_withQ_intr) の４つの状態で、待ち合わせ中端末からの onhook 信号受信に対する動作が個別に記述されている。これに対し、図１の記述では１個所ですんでいる。

また、図２では、condition で示される条件成立の場合分けにより遷移先の状態が異なる。このため、実際のＳＤＬ記述では、遷移先の状態を決定する手順をフローチャートとして記述する必要がある。これに対し、図１では、遷移先の状態解析はパターンマッチに集約され、形式的なパタン遷移規則の記述のみでよいという利点がある。

５．マッチング機能を利用する利点

(1)動作仕様のコンパクトな記述

マッチング機能により、複数の遷移規則が１つのパタン遷移規則によって表現可能となり、簡潔な記述が可能となっている。

(2)状態記述の意味理解性向上

従来、状態は全て異なる文字定数により名前付けされている。そのため、システムが大規模になって状態数が多くなってくると、略称や数字が多用され、状態の意味も理解しにくくなっていた。

図１の例では、状態を文字定数（プリミティブとして理解容易なもの）と変数を要素とするリストで表現した。そのため、独立性の高い状態構成要素は、リストの要素として個別に表現でき、理解性に優れている。また、リストは極めて柔軟な表現が可能であるので、必要性に応じた自在な表現が可能である。

(3)試験容易化の可能性

図１に示した記述に基づき、 Prolog による到達可能解析システムの生成が可能である。また、イリーガル状態への到達不可能性チェック（試験・検証の一手法）を、リスト表現に関する到達不可能性判定条件の十分条件として規定できる可能性がある。

６．おわりに

本稿では、マッチング機能を利用した仕様記述手法を提案した。本記述法の特長は以下のようにまとめられる。

(1)動作仕様の簡潔な記述が可能である。

(2)状態の意味理解容易化が可能である。

今後は、本記述手法を基本とした試験・検証手法について検討を進める予定である。

また本稿では、他記述法との比較のため通信システム仕様記述に話題を絞ったが、リーダーズアンドライターズ問題に代表される、不特定多数の装置からの要求信号の調停動作等、幅広い適用が考えられる。

［参考文献］

[1] Ginsparg and Gordon,"Automatic Programming of communications Software Via Nonprocedural Descriptions", IEEE trans. on Comm.,vol.com-30,no.6,JUNE,1982

[2] Imase,Hirakawa,et al,"Functional Programming Language for Switching Description and Its Hardware Architecture", ISS'84,13C2

[3] CCITT,"Functional Specification and Description Language (SDL)", Recommendations Z.100-Z.104 (1984)

<beginig state>	<input signal>	<next state>	<output signal>
[[talk X]..]	income(Y)	[[intr Y talk X]..]	
[[busy X]..]	income(Y)	[[intr Y busy X]..]	
[[intr X talk Y]..]	reject	[[talk Y]..]	reject(X)
[[intr X busy Y]..]	reject	[[busy Y]..]	reject(X)
[[intr X talk Y]..]	accept	[[talk X]..[wait Y]]	accept(X)
[[intr X busy Y]..]	accept	[[talk X]..]	accept(X)
[[talk X][wait Y]..]	roll	[[talk Y]..[wait X]]	
[[talk X][busy Y]..]	roll	[[busy Y]..[wait X]]	
[[busy X][wait Y]..]	roll	[[talk Y]..]	
[[busy X][busy Y]..]	roll	[[busy Y]..]	
[[intr Z talk X][wait Y]..]	roll	[[intr Z talk Y]..[wait X]]	
[[intr Z talk X][busy Y]..]	roll	[[intr Z busy Y]..[wait X]]	
[[intr Z busy X][wait Y]..]	roll	[[intr Z talk Y]..]	
[[intr Z busy X][busy Y]..]	roll	[[intr Z busy Y]..]	
[[talk X]..]	onhook(X)	[[busy X]..]	
[[intr X W Y]..]	onhook(X)	[[W Y]..]	
[[intr Z talk X]..]	onhook(X)	[[intr Z busy X]..]	
[..[wait X]..]	onhook(X)	[..[busy X]..]	
[X..]	onhook	[onhook X..]	
[onhook [intr X W Y]..]	λ	[onhook [W Y]..]	reject(X)
[onhook [talk X]..]	λ	[onhook ..]	onhook(X)
[onhook [busy X]..]	λ	[onhook ..]	
[onhook [wait X]..]	λ	[onhook ..]	onhook(X)
[onhook]	λ	idle	
Resource Blocking			
[[talk X]..]	income(Y)	[[talk X]..]	reject(X)
[[busy X]..]	income(Y)	[[busy X]..]	reject(X)

図１．キャッチホンサービスの記述例

<begining state>	<input signal>	<condition>	<next state>	<output signal>
talk_withQ	onhook(X)	condition1	busy_withQ	
		condition2	talk_withQ	
busy_withQ	onhook(X)	condition3	busy_withQ	
talk_withQ_intr	onhook(X)	condition4	talk_withQ	
		condition5	talk_withQ_intr	
		condition6	busy_withQ_intr	
busy_withQ_intr	onhook(X)	condition7	busy_withQ	
		condition8	busy_withQ_intr	

図２．従来記法による記述イメージ（一部）