数理解析研究所講究録
第591巻 1986年 183-193

「ステートマシン・ネットワークの関数による動作記述」における「満足」包含関係について

日本アイ・ビー・エム株式会社
山内長承 (Nagatsugu Yamanouchi)

1. はじめに

本報告は電子通信学会「オートマトンと言語」技術研究報告[1],[2]に続くものである。既発表分の要旨を整理すると

＊ステートマシンのネットワークの動作の解析を行なうモデルとツールを考える。空間的な接続関係を記述する構造モデルの上に，時間的な動作を記述する動作モデルを重ね合わせる。但し構造は時間的に不変とする。

＊構造モデルはモジュールのネットワークをグラフによって表現する。ネットワークをひとつのモジュールとする階層的な取扱いを許す。

ネットワークは要素モジュールを節，接続線を弧とする有向グラフとする。接続線には信号の方向によって向きを付けるものとする。モジュールには入力ポートと出力ポートが

あり，ラベルが付けられる．

モジュールからネットワークを合成する結合操作として，直列接続（composition）とループ接続（loop）の2種，および接続線を決める名札付替え（relabelling）を定義する。これらの操作によって任意のネットワークを生成できることを示し（Generator Lemma），更に標準形式をユニークに定義でき，異なる結合手順で作られた2つのネットワークが等しいか否かを，標準形式に変換することによって決めることができることを示した。（Normal Form Lemma）

＊モジュールの動作記述は次のように与える。モジュールの各接続線について入出力事象の列（Sequence）を与え，入力事象列から出力事象列への関数を動作モデルの核とする。非決定的動作をもつモジュールの場合は列の集合から列の集合への関数へ拡張してモデルする．[4]

この関数による記述に，入出力事象間のタイミングに依存する動作の記述を加える。入出力事象間のタイミングを入力に比べた出力のタイミングと，出力に比べた入力のタイミングの2種類に分ける。入力に比べて出力がいつ生成されるか（関数制約，出力がどの入力より早いか遅いか）は入出力関数に対して実際的な解釈を与えて導出できることを示した。他方，出力に比べて入力が早く到着しすぎる場合の記

述（定義域制約）は，タイミング関係式として 出力関数に併せて与える。 この早過ぎる入力の場合，モジュールの動作は矛盾状態に陥るものとし，その記述として入力に対して T（inconsistent）という出力を生成するものとする。

結合操作の動作上の意味は，直列結合は関数の合成，ループ結合は関数の最小不動点によって与えられ，名札付替えは動作上意味を持たない。 定義域制約は直列，ループ結合ともに各経路（入力ポート・出力ポートの組に対して定められる経由線の組み合わせ）の上での最大の先行入力数として求められる。 関数制約は入出力関数から求められる。

このようにして定義した結合操作の動作上の意味について，構造モデル同様，標準形式への変換手続きに必要な等式群が成立する。

2. 満足関係

今迄の枠組では，2つのネットワークが等しいことを示せるが，これだけでは不十分なことがある。 一般にネットワークの動作記述を要素のステートマシンの動作から合成し，その合成した系の動作記述 B_c と設計の意図の記述 B_s とを比較検討するのであるが，両者が一致することは必ずしも必要でなく，合成した系 B_c が設計 B_s を「満足する」か否か

を知りたい場合が多い。　この満足関係について検討する。

満足関係を次のように定義する。　2つのモジュール動作記述 B_c, B_s について、B_c が B_s を満足する ($B_c \sqsubseteq B_s$) とは

B_c の定義域制約 dc_c が B_s の定義域制約 dc_s より小さい、すなわち　$dc_c \subseteq dc_s$,　(1)

かつ

dc_s を満足するすべての入力列 X に対して、B_c の関数の出力 $f_c(X)$ は B_s の出力 $f_s(X)$ に等しい、すなわち

$f_c(X) = f_s(X)$。　(2)

条件(1)の記法は混乱を生ずる恐れがあるので次の注意を添える。　定義域制約の記法 dc は、制約の集合として表わしているので、定義域制約に抵触しない事象列の集合の包含関係は dc のそれと逆になる。　$dc_c \subseteq dc_s$ は S の方がより制約が多いことを示しており、dc_s に抵触しない事象列の集合は dc_c に抵触しない列集合に含まれる。

満足関係は次のような考察に基いて定義した。　モジュールの外界との係わり方をみると、合成モジュール C は、入力については、設計記述 S において意味ある出力を生成する入力をすべて受付けられなければならず、かつそれらの入力に対して S と同じ出力を生成しなければならない。　その他の

部分の入力に対しては設計記述Sと同じ出力を生成しなくてもよいものとする。 ここで意味ある出力を生成する入力とは，出力丁を生ぜしめない入力と解釈するのが現実的であり，dc_s を満足する入出力事象列である。 従って合成モジュールの定義域制約 dc_c は設計記述の制約 dc_s より緩やかでなければならない。

3. 満足関係の伝播

満足関係と結合操作について次の関係が成立する。 モジュール u_c, u_s に対して満足関係 $u_c < u_s$ が成立つ時，

[補題1]

モジュールひとの直列結合 $u_c \otimes ひ$, $u_s \otimes ひ$ が存在すれば

$$u_c \otimes ひ \quad < \quad u_s \otimes ひ .$$

同様に，$ひ \otimes u_c$, $ひ \otimes u_s$ が存在すれば

$$ひ \otimes u_c \quad < \quad ひ \otimes u_s .$$

[補題2]

ループ $u_c!$, $u_s!$ が存在すれば

$$u_c! \quad < \quad u_s! .$$

(証明の概略)

いずれも結合演算の定義域制約に対する意味[2]から，ほぼ

自明である。 満足関係の定義から u_c と u_s は入出力ポートについて同じ構造をしており，$u_c \otimes v$，$u_s \otimes v$ の経路は同じ集合となる。 各信号経路について定義域制約を決めるバッファ容量は，u_c（又は u_s）と v の対応する経路のバッファ容量 $C(u_c)$，$C(v)$ の小さい方 $\min(C(u_c), C(v))$ をとるが，満足関係 $u_c < u_s$ によってすべての経路について $C(u_c) \leq C(u_s)$ であるので，

$$\min(C(u_c), C(v)) \leq \min(C(u_s), C(v))$$

が成立つ．

4. モデルの応用例

今まで検討してきたモデルを現実に添った例に適用してみる。 ここで取上げるのは計算機網のオルタネーティングビット・プロトコール[5]である。

システムは，ネットワークと2人のユーザX，Yから成りネットワークは，信頼できない通信回線 mab，mba と，その両端で通信を制御する2つのプロセス A，B から成る。（図1） ユーザXがネットワークを介して送信するメッセージが，脱落・重複することなく，かつ正しい順序でYに到着することが検証したい。

ユーザXはメッセージのシーケンス $X_0 X_1 X_2 \cdots$ をプロ

セスAに渡す。 Aは図2に示す状態遷移をし，A_0 に乗せて X_0 を，A_1 に乗せて X_1 を，次の A_0 に乗せて X_3 を，順次送信してゆく。 各送信の間に必ずBからの返答を要求し，A_0 を送信した後は B_0 を，A_1 を送信した後は B_1 を要求する。 A_0 を送信した後 B_1 が返ってきた場合や受信エラーの場合（B_0 か B_1 かも分からない），A_0 を送信し続ける。 プロセスBも同様に，まず A_0 を受信するとその上に乗っているメッセージをユーザに渡し，B_0 を送信する。次に A_1 を受信することを期待し，もし A_0 または受信エラーであると B_0 を送信し続ける。

プロセスAの動作はたとえば入力

$X_0\ B_1\ B_1\ B_1\ B_0\ X_1\ B_0\ B_0\ B_1\ X_2\ \cdots$

に対して出力

$A_0\ A_0\ A_0\ A_0\ A_1\ A_1\ A_1\ A_0\ \cdots$

を生成する。 プロセスBについても上記の動作に対して，入力

$Er\ Er\ Er\ A_0\ Er\ Er\ A_1\ Er\ \cdots$

に対して出力

$B_1\ B_1\ B_1\ Y_0\ B_0\ B_0\ B_0\ Y_1\ B_1\ B_1\ \cdots$

というパターンがある。 但し Er は受信エラーを示す。

通信路 mab，mba は次のような動作パターンを持って

いる。 すなわち mab はエラーを生じ，入力

$A_0\ A_0\ A_0\ A_0\ A_1\ A_1\ A_1\ A_0 \ldots$

に対して出力

$E_r\ E_r\ E_r\ A_0\ E_r\ E_r\ A_1\ E_r \ldots$

を生じる。 他方，mba は幸運にもエラーがなく，入力

$B_1\ B_1\ B_1\ B_0\ B_0\ B_0\ B_1\ B_1 \ldots$

に対して出力

$B_1\ B_1\ B_1\ B_0\ B_0\ B_0\ B_1\ B_1 \ldots$

を生成する。

これらを結合して図1のシステム

$!(((A \otimes mab) \otimes B) \otimes mba)$

を作ると，上で使った動作パターンは一つのループ解になっている。 より本格的に計算するにはツールが不足するが，たとえば次のような記法を使って表現することができる。

<factor> ::= <event> | (<sequence>)

<term> ::= <factor> | <factor>$^{<count>}$

<sequence> ::= <term> | <sequence> + <term>.

意味は正規表現と同様に

+ : 二つの要素のいずれかを選ぶ，

F^c : Fをc回繰返す．

とする。 繰返し回数と事象をパラメータ化することによっ

て，複数の場合を1つの表現にまとめられる。

プロセスAの動作は

$$(X_i\,(B_{i\oplus1}+Er)^{k_i}\,B_{i\oplus2})^i \longrightarrow ((A_{i\oplus0})^{k_i+1})^i$$

と書け，Bの動作は

$$((A_{i\oplus0}+Er)^{k_i}\,A_{i\oplus0})^i \longrightarrow ((B_{i\oplus1})^{k_i}\,Y_i\,B_{i\oplus2})^i$$

と書ける。通信路 mab，mbaは，任意のj番目の事象U_jを含む入力$(U_j)^j$に対して，出力$(U_j+Er)^j$を生じる。

直列結合 $(((A \otimes mab)\otimes B)\otimes mba)$ の動作には

$$(X_i\,(B_{i\oplus1}+Er)^{k_i}\,B_{i\oplus2})^i \longrightarrow ((B_{i\oplus1})^{k_i}\,Y_i\,B_{i\oplus2})^i$$

を含む。これをループにすると

$$((B_{i\oplus1})^{k_i}\,B_{i\oplus2})^i$$

がループ解となり，対応する外部入出力は

$$(X_i)^i \longrightarrow (Y_i)^i$$

となり，データの脱落や順序の混乱のないことが示せる。停止性（liveness）はこのモデルでは k_i が有界であることに対応するが，これは通信路に対する仮定をおくことによって導ける。

(注) $i\oplus1$ は modulo 2 の加算を表わす。

5. まとめ

本報告では満足関係を定義し考察した。満足関係は結合操作について伝播するので，扱いは容易になると考えられる。後半では動作の簡単な解析例として，オルタネーティングビットプロトコールを取上げ，実際の解析には効率のよい表記ツールが必要なことを示した。

6. 参考文献

[1] 山内長承："ステートマシン・ネットワークの関数による動作記述（1）——記述系の定義と結合操作"，信学技報 AL 85－15.

[2] 山内長承："ステートマシン・ネットワークの関数による動作記述（2）—— クロス・タイミング，非決定的動作などの問題点について"，信学技報 AL 85－22.

[3] 山内長承："ブロック・アッカーマン問題に対する一考察"，情処 30回全国大会，pp 369－370.

[4] Plotkin, G.D.: "A Powerdomain Construction." SIAM J of Computers 5(3), 1976.

[5] Bartlett, K.A. and Scantlebury, R.A.: "A note on reliable full-duplex transmission over half-duplex links." CACM, 12(5) 1969.

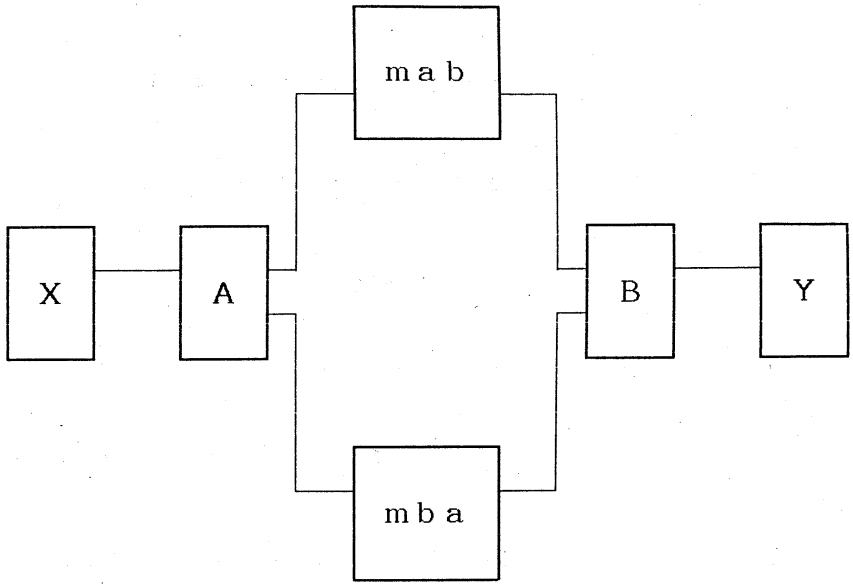

Fig. 1. Alternating Bit Protocol

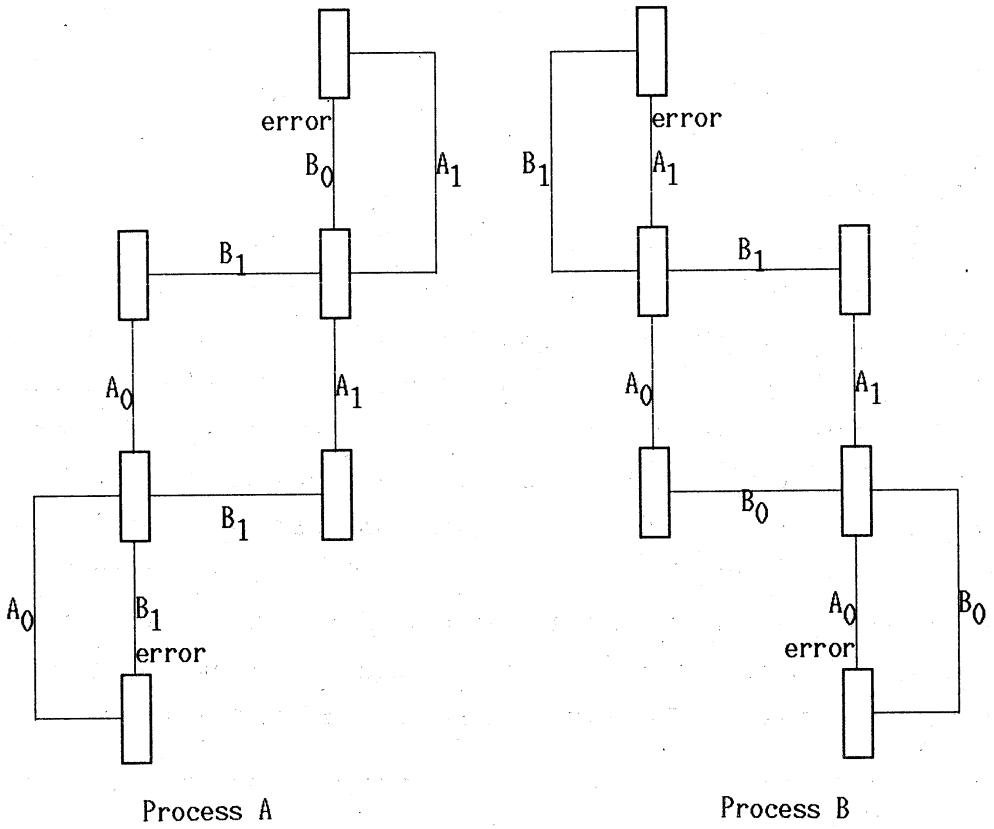

Fig. 2. Process A and B.