4G−2

ペトリネットのための代数的操作システムとその応用

川本 真一　伊藤 貴康
東北大学 工学部 情報工学科

1. はじめに

ペトリネットに対する代数的体系としては [1] があるが、これはネットそのものに対する体系であってプロセスに対する体系にはなっていない。そこで、ペトリネットによって表現されたプロセスを代数的に扱うことのできるネット代数の可能性について検討し、ペトリネットのための代数的操作システムを提案する。本研究ではネットそのものの操作(NO)の他に、プロセス操作に用いられるオペレーションをネットレベルに導入し(PO)、これとプリミティブネット(PN)から代数的な性質を持つ代数ネットANを定義し、これをもとに代数的操作システムを構成した。POはOccam[3][4]や、CCSP[2]などのプロセス構成子に相当するものである。代数ネットANの応用として、Occamプロセスを代数ネットANで表現し、プロセスの等価性を保存したネットの変換式を用いて、Occamに対するANを簡単化することができることを示す。

2. ペトリネットオペレーションとプロセスオペレーション

プロセスをネットによって表現するという観点から、基本となるネットをラベル付きプレーストランジションネットとする。ただし、プレースの容量はすべて∞とする。以後このクラスをUCネット(Unbounded Capacity Net)と呼ぶことにする。

Definition 1 UCネット(UCN)
UCネットとは次の6項組である。

$$N = (P, T, F, W, M_0, L)$$

次にUCネットNを操作する基本的なオペレーションを定義する。ただし、UCNの集合を$\mathrm{N_U}$とし整数の集合をZとする。

Definition 2 ネットオペレーション(NO)

- NO1. **Superposition** $SP : \mathrm{N_U} \times \mathrm{N_U} \rightarrow \mathrm{N_U}$
  2つの別々なネットを重ね合せて1つのネットを作る。
- NO2. **Place Merging** $PM : \mathrm{N_U} \times P \rightarrow \mathrm{N_U}$
  指定されたプレースの集合をすべてマージして1つのプレースにする。
- NO3. **Transition Merging** $TM : \mathrm{N_U} \times T \rightarrow \mathrm{N_U}$
  指定されたトランジションの集合をすべてマージして1つのトランジションにする
- NO4. **Token Increment** $TI : \mathrm{N_U} \times P \times \mathrm{Z} \rightarrow \mathrm{N_U}$
  プレースpのトークンをZ個だけ増やす。
- NO5. **Transition Elimination** $PE : \mathrm{N_U} \times T \rightarrow \mathrm{N_U}$
  指定されたトランジションを取り去る。

NOはネットそのものを操作するオペレーションであるが、次にネットをプロセスのように操作するオペレーションPOを定義する。ただし、ネットNに関して、${}^{\cdot}N, N^{\cdot}$をそれぞれ、ネットの入出力プレースの集合を表すものとし、$L(s)$は集合sの全要素に依存したラベルを表し、$N_1, N_2 \in \mathrm{N_U}$とする。

Definition 3. プロセスオペレーション(PO)

- PO1. **Sequential Composition** (;)
  $N_1; N_2 = SP(\{PM(SP(m_1, m_2), m_1^{\cdot} \cup {}^{\cdot}m_2) \mid m_1 \in dec(N_1)\ ,\ m_2 \in dec(N_2)\})$
- PO2. **Nondeterministic Choice** (+)
  $N_1 + N_2 = SP(\{PM(PM(SP(m_1, m_2), {}^{\cdot}m_1 \cup {}^{\cdot}m_2), m_1^{\cdot} \cup m_2^{\cdot}) \mid m_1 \in dec(N_1)\ ,\ m_2 \in dec(N_2)\})$
- PO3. **Parallel Composition** (|)
  $N_1 | N_2 = SP(N_1, N_2)$
- PO4. **Iteration** (*)
  $*N_1 = SP(\{TI(PM(m, m^{\cdot} \cup {}^{\cdot}m), L(m^{\cdot} \cup {}^{\cdot}m\}), +1) \mid m \in dec(N_1)\})$

ただし、decは[2]において用いられた概念を修正したものであり、ネットの並列に実行される要素を逐次的に動作する複数のネットに分解する。その定義は次のとおりである。

Definition 4. Decomposition 関数 (dec)

$$dec(N) = \begin{cases} \{N\} & (\text{if } N \text{が} \mid \text{を含まない}) \\ \{M_1; M_2 \mid M_i \in dec(N_i)\} & (\text{if } N = N_1; N_2) \\ \{M_1 + M_2 \mid M_i \in dec(N_i)\} & (\text{if } N = N_1 + N_2) \\ \{*(M_1; M_2) \mid M_i \in dec(N_1)\} & (\text{if } N = *N_1) \\ dec(N_1) \cup dec(N_2) & (\text{if } n = N_1 | N_2) \end{cases}$$

3. 代数ネットとペトリネットのための代数的操作システム

3.1 代数ネットとその性質

プロセスは一般に基本要素に演算を適用することによって構成される。このプロセスの基本要素に対応するネットが次に定義するプリミティブネット(PN)である。

Definition 5. プリミティブネット(PN)
PNは右図のようなネットである。

ただし、xはトランジションの識別子であり、pi_x, po_xはxで表されたトランジションの入力プレースと出力プレースの識別子であるとする。また、Lはラベルを表すものとする。PNはその構造からトランジションの識別子のみによって表現することができるので、以後は各PNをトランジションの識別子によって表す。

次に、基本要素PNと、プロセスオペレーションPOから代数ネットANというUCNのサブクラスを構成する。ただし、トランジションとプレースが空(ϕ)であるUCNをN_ϕで表す。

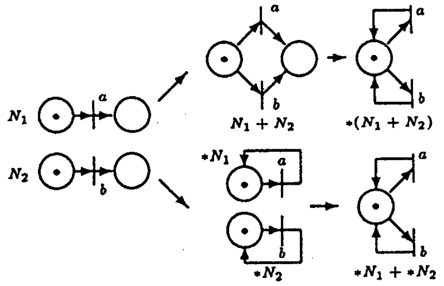
Definition 6. 代数ネット(AN)

- (i) N_ϕはANである。
- (ii) PNはANである。
- (iii) ANにPO(; , + , | , *)を適用した結果もANである。

POの2項演算子 ; , + , | は、ANに対して以下のようにN_ϕを単位元とし、結合則が成り立つ。また + , | については可換則、*については分配則などが成り立つ。この性質より、PNとPOより構成されるネットを代数ネットと名付けた。

$N; N_\phi = N_\phi; N = N$　　$N + N_\phi = N_\phi + N = N$
$N | N_\phi = N_\phi | N = N$　　$(N_1; N_2); N_3 = N_1; (N_2; N_3)$
$(N_1 + N_2) + N_3 = N_1 + (N_2 + N_3)$　　$(N_1 | N_2) | N_3 = N_1 | (N_2 | N_3)$
$N_1 + N_2 = N_2 + N_1$　　$N_1 | N_2 = N_2 | N_1$
$*(N_1 + N_2) = *N_1 + *N_2$

<例>



これは一例であるが、一般の代数ネットに対しても$*(N_1 + N_2) = *N_1 + *N_2$が成り立つ。

Algebraic manipulation system for Petri nets and its applications
Shinichi Kawamoto , Takayasu Ito
Tohoku University

3.2 代数ネットによる Occam プロセスの表現

Occam プロセス Op を代数ネット AN に変換する関数操作は、ネット構造を与える操作$\mathcal{N}$と、同期機構を与える操作 $Sync, CE$ より構成される。これらの定義は以下のとおりである。

Definition 7. ネット構造化 $\mathcal{N}$: Op → N

1. Op がプリミティブプロセスの場合
   $\mathcal{N}(Op) = t_i$ (t_iは PN であり、$L(t_i) = Op$ とする.)
2. Op が SEQ プロセスの場合 (SEQ $P_1, P_2, \cdots, P_n$)
   $\mathcal{N}(Op) = \mathcal{N}(P_1); \mathcal{N}(P_2); \cdots; \mathcal{N}(P_n)$
3. Op が ALT プロセスの場合 (ALT $g_1P_1, \cdots, g_nP_n$)
   $\mathcal{N}(Op) = \mathcal{N}(g_1); \mathcal{N}(P_1) + \cdots + \mathcal{N}(g_n); \mathcal{N}(P_n)$
4. Op が IF プロセスの場合 (IF b_1P_1, false$P_2, \cdots, b_nP_n$)
   $\mathcal{N}(Op) = \mathcal{N}(P_1) + \mathcal{N}(P_3) + \cdots + \mathcal{N}(P_n)$
5. Op が WHILE プロセスの場合 (WHILE bP)
   $\mathcal{N}(Op) = *\mathcal{N}(P)$
6. Op が PAR プロセスの場合 (PAR $P_1, P_2, \cdots, P_n$)
   $\mathcal{N}(Op) = \mathcal{N}(P_1)|\mathcal{N}(P_2)|\cdots|\mathcal{N}(P_n)$

ただし、1 において、Op に出現するすべてのプリミティブプロセスに異なった PN を割り当てるものとする。

Definition 8. 同期化関数 $Sync$: N → N

$$Sync(N_o) = SP(\{TM(N_o, iop) \mid iop \in IOP(N_o)\})$$

ただし、$IOP(N_o)$ は、N_oに現れチャネルが一致する入出力プロセス t_i, t_o の組 $\{t_i, t_o\}$ の集合であり、$N_o = \mathcal{N}(Op)$ とする。

Definition 9. 入出力削除 CE : N × T → N

$$CE(N_s, IC) = TE(TE(\cdots TE(N_s, t_1), \cdots, t_{n-1}), t_n)$$

ただし、ICは外部と通信を行うものを除いたすべての通信プロセスの集合であり、$IC = \{t_1, \cdots, t_n\}$ とする。また、$N_s = Sync(N_o)$ とする。

$Sync$ は入出力プロセスに相当する PN をマージして通信が同期的に行なわれるようにするものであり、CEはマージされずに残っている (実行されることのない) 入出力プロセスに相当する PN を取り除く操作である。$\mathcal{N}, Sync, CE$ を用いると Occam プロセス Op のネット表現は次のように表される。

$$CE(Sync(\mathcal{N}(Op)), IC)$$

次に、[3] によって示された Occam プロセスの等価変換公式の両辺を、$\mathcal{N}$によって代数ネット AN に変換したものを示す。これは、Occam プロセスの等価性を保存するネットの変換公式となる。ただし、⇔ は相互に変換可能であることを示すものとする。また、異なった識別子によって表される 2 つのネット N_i, N_j が、同じ構造を持ち、トランジションのラベルが一致するとき $N_i =_f N_j$ と書くものとすれば、N_i' $(i = 1, 2, 3)$ は $N_i =_f N_i'$を満たすものとする。

Op の等価性を保存した AN の変換式 (PET)

PET1.	$N_1 + N_2 \Leftrightarrow N_1 \Leftrightarrow N_2$ (if $N_1 = N_2$)
PET2.	$N + t \Leftrightarrow N$ (if $L(t) = \mathrm{STOP}$)
PET3.	$t_1; N \Leftrightarrow N; t_2 \Leftrightarrow N$ (if $L(t_1) = L(t_2) = \mathrm{SKIP}$)
PET4.	$(N_1 + N_2); N_3 \Leftrightarrow (N_1; N_3) + (N_2; N_3')$
PET5.	$(N_1 + N_2)\|N_3 \Leftrightarrow (N_1\|N_3) + (N_2\|N_3')$
PET6.	$N_1; (N_2 + N_3) \Leftrightarrow (N_1; N_2) + (N_1'; N_3)$
PET7.	$N_1\|t \Leftrightarrow N_1$ (if L(t) = SKIP)
PET8.	$*N_1 \Leftrightarrow (N_1; *N_1') + t$ (if $L(t) = \mathrm{SKIP}$)
PET9.	$*N_1 \Leftrightarrow *(N_1 + *t)$ ($L(t) = \mathrm{SKIP}$)
PET10.	$*N_1 \Leftrightarrow *N_1 + t$ (if $L(t) = \mathrm{SKIP}$)
PET11.	$*(N_1; N_2) \Leftrightarrow (N_1; *(N_2; N_1'); N_2') + t_1$ (if $L(t_1) = \mathrm{SKIP}$)

例として、pipeline を実現する Occam プログラム [4] を考える。ただし、型宣言は省略するものとする

- (list) pipeline をシュミュレーションする Occam プログラム -

```
-- FEED process --
PROC feed ()
  SEQ
    more := TRUE
    WHILE more
      SEQ
        input ? next
        IF
          next < 0
            SEQ
              more := FALSE
              pipe[0] ! next
          next >= 0
            pipe[0] ! next
:
```

```
-- PIPELINE process --
PROC pipeline ()
  PAR i = 0 FOR n
    SEQ
      more := TRUE
      WHILE more
        SEQ
          pipe[i] ? next
          IF
            next < 0
              SEQ
                more := FALSE
                pipe[i+1]!next
            next >= 0
              SEQ
                processing()
                pipe[i+1]!next
:
```

```
-- BLEED process --
PROC bleed ()
  SEQ
    more := TRUE
    WHILE more
      SEQ
        pipe[n] ? next
        IF
          next < 0
            more := FALSE
          next >= 0
            output ! next
:
```

```
-- MAIN process --
PAR
  feed ()
  pipeline ()
  bleed ()
:
```

$feed, bleed, pipeline$ の各プロセスを$\mathcal{N}$によって変換すると、$\mathcal{N}(feed)$ には PET3. と PET5. が適用可能であり、$\mathcal{N}(pipeline)$ には PET5. がn 回適用可能であり、それぞれ簡単化できる。これら簡単化されたものを用いて $Main$ プロセスを構成し、$Sync$ と CE を適用することによって得られる代数ネット AN を図示すると図 1 のようになる。

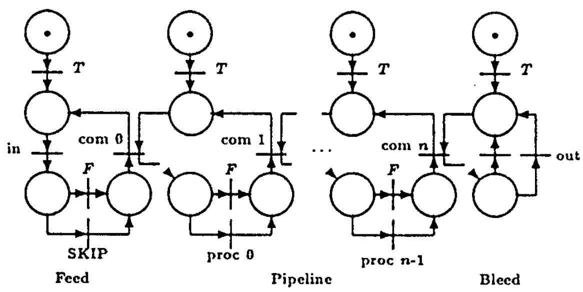


図1

4. まとめ

プロセスオペレーションによってネットを代数的に組み立てていくための枠組である代数ネット AN を定義し、AN を用いて Occam プロセスのモデリングと簡単化を行なった。CCSP などの他のプロセスモデルも PO との対応関係から容易に AN によって表現できるものと考えられる。

参考文献

[1] V.E. Kotov , *An algebra for parallelism based on petri nets*, LNCS 64 1978

[2] E.R. Olderog , *Operational Petri Net Semantics for CCSP* , LNCS 266 1987

[3] A.W.Roscoe and C.A.R Hoare , *The laws of occam programming*, TCS 60 1988

[4] D.Pountain and D.May, *Tutorial introduction to occam programming*, Blackwell Scientific Publications, 1987

本研究は科研費一般 (A)01420029 の一環として行われたものである。