LOTOS検証システム*Metis*II

1T−1

川口研治†、高橋薫††、白鳥則郎†††、野口正一†
†東北大学応用情報学研究センタ、††東北大学電気通信研究所
†††東北大学工学部

1. はじめに

通信プロトコルなどの厳密で曖昧さのない仕様を記述するために形式記述技法(FDT)が研究されている。数学的定義に基づくモデルを持つFDTを利用することにより、仕様の記述内容の形式的な取り扱いが可能となる。その内の一つに仕様の検証という問題がある。本稿ではLOTOS[1]により記述された二つの仕様間の等価性の検証を行うシステム、*Metis*IIについて述べる。

等価性検証の結果が等価となるようにするために、否等価の場合はもとの仕様を修正する必要がある。しかし単に等価性を判定しただけでは修正のための手がかりは数学的モデルであるLTS上にしか与えられない[4]。そこで*Metis*IIでは等価性判定を行うだけでなく、特に検証結果が否等価となった場合に、その原因となったLTS上の部分とそれのテキスト表現上の位置との対応を取ることにより修正の際のてがかりを与えることも重要な目的としている。

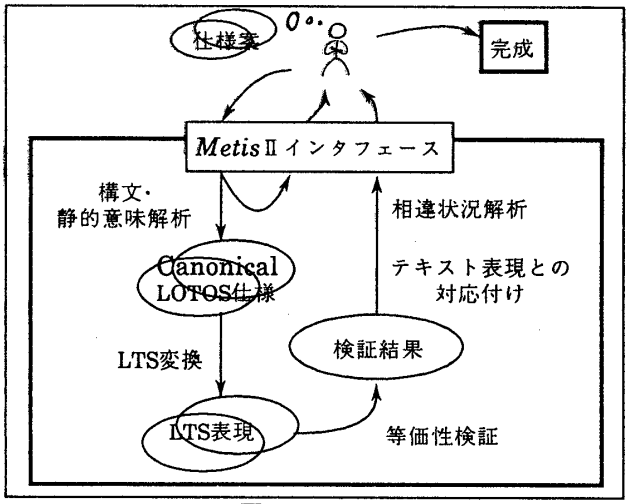

図-1:*Metis*II

2. *Metis*II

本システムは図-1に示すような過程で仕様の検証とその修正のための支援を行う。利用者はまず仕様をLOTOSにより記述し、システムに与える。*Metis*IIは構文解析と静的意味解析を行い文法的に正しいもののみを受け付ける。その後動的意味解析を行いLabeled Transition System(LTS)を導出する。LTSが有限であれば検証を続けることができ、等価か否等価かを決定する。否等価の場合にはその原因となる遷移あるいは状態を特定してその理由を示し、さらにそれらがもとのテキスト表現中のどの部分の記述により構成されたものなのかを明示する。以下では等価性検証とテキスト表現との対応付けについて説明を行う。

3. 等価性検証

LTSが有限である場合については状態の同値類分割を繰り返すことにより等価性を判定するのが一般的な方法である[2][3]。ここでLTSを状態の集合S、初期状態s_0、アクションの集合A及び遷移関係の集合Tで表し、さらにRをSの上の二項関係とし、

$F(R)=\{(p,q)|p,q\in S$ and for every $\mu\in(A-\{i\})\cup\{\epsilon\}$
(i)if $p=\mu=>p'$ then $(\exists q':q=\mu=>q'$ and $p'Rq')$
(ii)if $q=\mu=>q''$ then $(\exists p'':p=\mu=>p''$ and $p''Rq'')\}$

とする時、二項関係$\approx_k,k\geqq 0$を以下のように再帰的に与える。

(a)$\approx_0=S\times S$
(b)$\approx_{k+1}=F(\approx_k),k\geqq 0$

もしp$\approx_k$qならpとqはk-弱bisimulation等価であると言う。もしすべてのk≧0においてp$\approx_k$qならpとqは弱bisimulation等価であると言い、p≈qと書く。つまり$\approx=\cap\approx_k,k\geqq 0$である。

一方、二項関係$\approx_k$は同値関係なので、各$\approx_k$に関する同値類の集合π_kを得ることができる。≈に関する同値類の集合πにおいて、二つのLTSの初期状態が同じ分割に属しているならば等価であり、そうでなければ等価でないということがいえる。

3.1. 原因となる状態

双方の初期状態が同じ同値類に属さなくなるのはそれらの生起可能イベントが一致しないか、あるいはその遷移先の状態に原因を求めることができる。双方の初期状態s_1とs_2が関係$\approx_n$であるが$\approx_{n+1}$でなくなるのは、$s_1=a=>s_1',\exists a\in(A-\{i\})\cup\{\epsilon\}$の時$s_2=a=>s_2'$となる$s_2'$がないかあるいは$s_1'$と$s_2'$が関係$\approx_n$にないからである。ここでaの遷移先で関係$\approx_n$になる状態がそれぞれ1個づつの場合それら$s_1'$と$s_2'$が関係≈であれば$s_1\approx s_2$となる。そこで今度は$s_1'$と$s_2'$が関係≈にならない原因を探ることになる。このように$s_1\approx s_2$かどうかの問題が、その遷移先の状態同士が関係≈にあるかどうかの問題へと移っていく。そして状態対が決定できなくなるとそれ以上先の遷移先状態は比較できないのでそこが

"*Metis*II,the LOTOS verification system."
Kenji KAWAGUCHI†, Kaoru TAKAHASHI††, Norio SHIRATORI†††, Shoichi NOGUCHI†
†Research Center for Applied Information Sciences, Tohoku University.
††Research Institute of Electrical Communication, Tohoku University.
†††The Faculty of Engineering, Tohoku University.

否等価の原因として示されることになる。

次にアルゴリズムを示す。

$Sys1=<S_1,A,T_1,s_{01}>$,$Sys2=<S_2,A,T_2,s_{02}>$ を任意のLTSとし、$S=S_1 \cup S_2$とする。Ncを原因となる状態対の集合とし、Nを状態対の集合とする。アルゴリズム終了時にNcが空であれば等価でありそうでなければ等価でない。

『表記法』

●各状態sからの生起可能イベントの集合。
$A(s)=\{a|(s,a,s')\in T\}$ □

●各状態からあるイベントによって遷移可能な状態の集合。$S(s,a)=\{s'|(s,a,s')\in T\}$ □

●状態の集合$s \subseteq S$とSの$\approx_k$による分割π_kにおいて、sの要素が属する同値類の名前の集合。
$Name(s,\pi_k)=\{name(B_i)|B_i\in\pi_k, p\in B_i, p\in s\}$
但し、$name(B_i)$はB_iのπ_k中でのユニークな名前を表わす。 □

『アルゴリズム』

1.$Nc=\phi$,$N=\{(s_{01},s_{02})\}$とする。

2.$N=\phi$なら終了。このときのNcが求めるもの。$N\neq\phi$なら$\exists(s_1,s_2)\in N$を選び、$N=N-\{(s_1,s_2)\}$とする。

3.もし$A(s_1)\neq A(s_2)$ならば$Nc=Nc\cup\{(s_1,s_2)\}$とし2へ。さもなくば4へ。

4.s_1,s_2が初めて別の同値類の要素になるπ_kを求める。もし$s_1\approx s_2$ならば2へ。

5.s_1,s_2が初めて別の同値類の要素になるπ_kの一つまえのπ_{k-1}において、各$a\in A(s_1)$毎に$L_a=Name(S(s_1,a),\pi_{k-1})$,$R_a=Name(S(s_2,a),\pi_{k-1})$とし、$L_a'=L_a-R_a$,$R_a'=R_a-L_a$とする。

6.$L_a'=R_a'$となるものは除外する。L_a',R_a'のうち少なくとも一方の要素数が1以外となるaが存在するならば、$Nc=Nc\cup\{(s_1,s_2)\}$とし2へ。さもなくば7へ。

7.$Bl\in L_a'$,$Br\in R_a'$の要素数が共に1個なら、各々の要素$s_{La}\in Bl$, $s_{Ra}\in Br$の状態対(s_{La},s_{Ra})を$N=N\cup\{(s_{La},s_{Ra})\}$として2へ。さもなくば$Nc=Nc\cup\{(s_1,s_2)\}$とし2へ。 □

4. テキスト表現との対応

LTS上のある状態を表わす動作式やイベントがテキスト上のどの部分の動作式によって構成されているのかがわかるようにするためには、初期動作式をどのような手順でどのように変化させていったのかがわかれば良い。そこで、ある状態が初期状態からどのような公理·推論規則の適用をうけてきたかと言う情報を保存しておくようにする。そして各状態を表わす動作式を初期動作式及びプロセス名とそれに適用してきた公理·推論規則のみで表現することを考える。

ここではそのような表現のことを履歴表現と呼び、その表記法·意味·構成法を定義することにより対応が取れるようにする。なお意味·構成法の詳細な定義は紙面の都合上省略する。

『表記法』

<状態>::=<基本動作式>
　　|(<状態>,<ルールlist>)
<基本動作式>::=init|<プロセス名>
<ルールlist>::=<ルール>
　　|<ルール>(<ルールlist>)
<ルール>::=pi(<状態>)|int(<状態>,<状態>)
　　|par(<状態>,…,<状態>)|ap|ex|en1|en2
　　|di1|di2|di3|ch1|ch2|pp|hi|sum □

<ルール>中で使用している記号は、一つ一つがそれぞれ各公理や推論規則に対応している。又、<状態>が指示点の出発点を与え、ルールlistがそこからの指示点の動かし方や動作式の範囲を与える。

4.1. 意味

履歴表現を解釈することにより、LTS上の各状態が元のテキスト表現上のどの部分から構成されるかを知るためのものである。各ルール毎に三つのことが定義される。すなわち1.基本動作式を表わす部分木上での指示点の動かし方。2.部分木の範囲の与え方。3.イベントの与え方。履歴表現の意味を解釈するにはまず1.に従い指示点を移動させた後、2·3に従いイベントや動作式の位置を得る。

4.2. 構成規則

これは各状態の履歴表現がどのようにして得られるのか、さらには全ての状態が必ず履歴表現によって表わすことができるのだろうかと言うことを説明するのに用いられる。

ある状態から新しい状態が導出されたときに成立した公理·推論規則を、それらが成立した逆の順に並べたものをルールlistとする。このルールlistを前の状態の履歴表現に追加することによって新しい状態の履歴表現を得ることにすれば各状態の履歴表現が手続き的に得られるようになる。又、新しい状態が導出される時には必ずルールlistが得られるので、初期状態が履歴表現されれば全ての状態が必ず履歴表現によって表わされることがわかる。このような、履歴表現にルールlistを追加する規則がこの構成規則である。ここでは初期状態を「init」とし、追加のための演算記号を「+」とする。規則は状態への追加とルール同士の合成の二種類により構成される。

5. おわりに

本システムは現在Sun-3上のUNIX環境中にCを用いて構築中である。

[参考文献]

[1]ISO : "LOTOS − A formal description technique based on the temporal ordering of observational behaviour",ISO 8807(Feb. 1989).

[2]Tommaso Bolognesi et al.: "Fundamental results for verification of observational equivalence:a Survey." In proc. of 7th IFIP symposium on protocol specification, testing, and verification (May 1987).

[3]神長裕明 他: "LOTOS仕様の等価性とその判定法", 信学会論文誌D-I Vol. J72-D-I No.5 (May 1989).

[4]川口研治 他: "LOTOS仕様の等価性検証システムの構築", 信学技報IN89-59 (July 1989).