SAL:LOTOS仕様の意味解析支援システム －実現方式－

4N－5

川口研治†、 佐藤淳††、 高橋薫†、 白鳥則郎†、 野口正一†
†東北大学電気通信研究所
††富士通カストマエンジニアリング

1. はじめに

システムの形式的な仕様記述を行うための技法の一つにLOTOS[1]がある。LOTOSにより記述された仕様の表す動作の確認を行うためには、その動的意味解釈が必要となる。本稿ではLOTOS仕様の動的意味解釈を支援するSAL(Semantic Analyzer of LOTOS)システムの実現方式について述べる。

2. 実現方式

SALシステムはLOTOS仕様からその意味モデルであるLabelled Transition System(LTS)を生成する。生成されたLTSは他の仕様解析、例えば試験系列の生成・仕様の等価性の検証等に利用することを目的としている。ここで生成されるLTSの各イベントには変数やガード、条件等式がついたままになっており、導出されたLTS(変数付きLTS)を利用する他のシステムがその必要に応じて変数の値を具体的に評価し、LTSを拡張することができるようにしてある。

2.1 全体の構成

SALシステムは次のことを行う2つのプログラムから構成されている。一つめはLOTOS仕様を読みこみ構文解析、静的意味解析及び平滑化を行う仕様解析フェーズであり、二つめは平滑化の済んだ仕様を読みこみ公理・推論規則を用いて変数付きLTSを生成する仕様展開フェーズである。

2.2 仕様解析フェーズ

ここでは構文解析、静的意味解析及び平滑化を行う。このフェーズの作成にはyacc&lexを利用し、静的意味解釈はyaccのアクションとして静的意味解析プログラムを用意しておき、これを呼び出すことによって行う。具体的には以下のことを行う。

[構文解析]

- コメントの除去
- 構文要素の分離と識別
- 識別子の定義オカレンスと適用オカレンスの検出
- 前置記法への変換

[静的意味解析]

- スコーピング規則を満足するかどうかの判定
  定義オカレンスを検出した際、そこのスコープ情報を付加して定義オカレンスのテーブルに格納する。一方適用オカレンスを検出した際に定義オカレンスのテーブルを検索し、識別子のオカレンス及びスコープ情報が一致しているかを判定する。
- 識別子のユニーク化
  識別子の定義オカレンスの出現回数により、識別子の後に整数値を付加しユニーク化する。また適用オカレンスにおいては、同一スコープのユニーク化した識別子の定義オカレンスを用いる。

2.3 仕様展開フェーズ

ここでは仕様の動的意味解釈を行う。仕様展開フェーズは、図-1に示される5つのモジュールから構成されている。平滑化仕様入力部においては仕様解析フェーズの出力を表の形に直し、LTS生成部に渡す。LTS生成部では遷移導出部を繰り返し呼び出して遷移関係を得て、仕様の動的意味である変数付きLTSを導出する。遷移導出部は各状態(動作式)での生起可能イベントとその遷移先状態を求めるために、動作式を解析し適当な公理・推論規則を選び適用する。イベントの値の解釈の際には、項解釈部が必要に応じて呼び出される。結果的に得られた変数付きLTSはLTS出力部を通して、ファイルに格納されたり、ユーザにグラフィック表示されたりする。

"Implementation method for the SAL system."
Kenji KAWAGUCHI†, Atushi SATO††, Kaoru TAKAHASHI†, Norio SHIRATORI†, Shoichi NOGUCHI†
†Research Institute of Electrical Communication, Tohoku University.
††Fujitsu customer engineering,Limited.

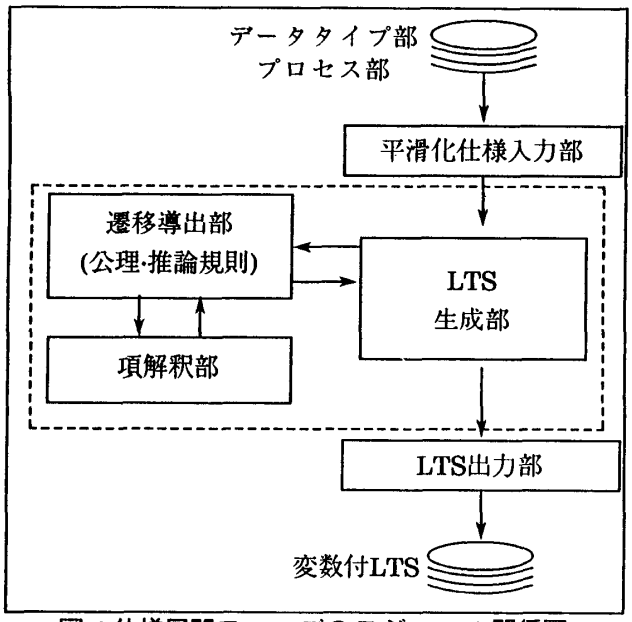

図-1 仕様展開フェーズのモジュール関係図

[平滑化仕様入力部] 仕様解析フェーズにより出力されたファイルを読みこむ。データタイプ部からはsorts(ソート)、opns(オペレーション)、eqns(等式)の部分を、又プロセス部からは各プロセス定義の識別子、フォーマルパラメータリスト、動作式の部分を抜き出して、それぞれ参照を行いやすくするために表のかたちにまとめる。

[LTS生成部] LTSは図-2に示すように、状態を表現するheader部と遷移関係を表現する遷移部によって表される。

まず初期プロセス定義の動作式をとりだしてheaderを作成し、その動作式を遷移導出部に渡して遷移関係のリストである遷移部を得る。それをheaderにつなげる一方、新しく導出された動作式に対してそれぞれheaderを作成する。ただし新しく得られた動作式と同じものが自分の祖先の状態にあれば、新しくheaderを作成せずにすでにある動作式のheaderを遷移先とし、その動作式については以後導出が繰り返されることがないようにする。新しく作成されたheaderは、自分の動作式に導出規則が適用される順番がくるまで待ち行列に入れられる。

以上のことを待ち行列からとりだしたheaderの動作式に対しても行う。待ち行列が空になった時点でLTSの生成は終了する。

[遷移導出部] 適用すべき公理·推論規則(ルール)の選択及び表現を行いやすくするために、動作式をその式中で最も優先順位の低いオペレータとその左右の部分の三つの部分にわけて扱う。sum、par、

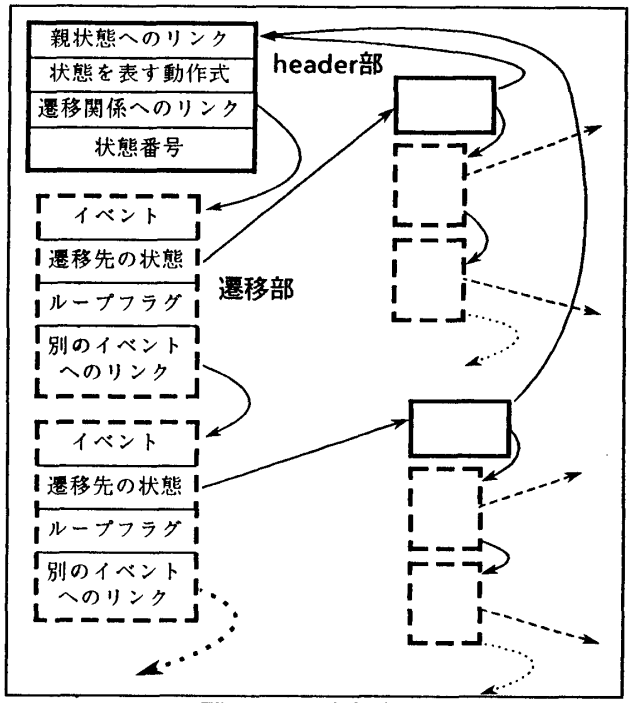

図-2 LTSの内部表現

hideの各動作式は二つの部分に分けられる。

ルールは各オペレータ毎にそれに関係するものをまとめて表現し、それぞれ独立に変更、追加が行えるようにする。これにより、仕様中で使われているオペレータに関するルールさえそろっていれば導出が可能となる。又、適用すべきルールの選択はオペレータをみることにより決定することができる。もし遷移が可能ならば図-2中の遷移部に示される形で遷移関係のリストが得られる。

ガードや項の値を知る必要があるときは項解釈部をよびだし、評価させる。

[項解釈部] データタイプの定義に基づき項の値の評価を行う。なお、現在の版では人間がこれを行なっている。

3. 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。変数付きLTSのグラフィック表示にはSunViewを利用している。なお、項の解釈は現在人間に判断を求めているが、この部分にTRSを組み込んで自動的に解釈させることを検討中である。

[参考文献]

[1] ISO : "LOTOS − A formal description technique based on the temporal ordering of observational behaviour",ISO 8807(Feb. 1989).
[2] 高橋:"SAL:LOTOS仕様の意味解釈支援システム",東北大学内部資料(1989).