

LOTOSの汎用的な中間言語

佐藤 嘉一(沖電気工業(株)コンピュータシステム開発本部)
 辻 宏郷(三菱電機(株)情報電子研究所)
 内山 光一((株)東芝 情報通信システム技術研究所)
 小野 昌秀(沖電気工業(株)コンピュータシステム開発本部)
 五ノ井 敏行(富士通(株)情報システム事業本部)
 田中 功一(三菱電機(株)情報電子研究所)
 藤田 朋生(日本電気(株)基本ソフトウェア開発本部)
 山中 顕次郎(日本電信電話(株)NTTソフトウェア研究所)
 大薮 和仁(電子技術総合研究所 情報アーキテクチャ部)

通信プロトコルOSIのための形式仕様記述言語LOTOSは国際規格ISO8807となっている。我々は、主にLOTOSの実用性に関する調査/研究を目的として、研究会を行ってきた。その活動内容は、LOTOSの勉強、実際のプロトコルのLOTOSによる記述実験、ASN.1からLOTOSの抽象データ型への変換の検討、“LOTOSの処理系”の検討及び設計などである。

本稿では、我々の設計しているLOTOS処理系の扱う中間言語の概要を紹介する。この中間言語は、LOTOSで記述された仕様を扱う様々なツールが、共通に扱うことのできるよう、構文解析、静的意味解析の結果を忠実に反映するものとして設計されている。

Design of General Purpose Intermediate Notation for LOTOS

Y.Sato(Ok Electric Industry Co.,Ltd.)
 H.Tsuji(Mitsubishi Electric Corporation)
 M.Uchiyama(Toshiba Corporation)
 M.Ono(Ok Electric Industry Co.,Ltd.)
 T.Gonoi(Fujitsu Limited)
 K.Tanaka(Mitsubishi Electric Corporation)
 T.Fujita(NEC Corporation)
 K.Yamanaka(Nippon Telegraph and Telephone Corporation)
 K.Ohmaki(Electrotechnical Laboratory)

LOTOS is a language to specify OSI protocols formally. LOTOS has been ISO 8807 since February 1989. We are organizing a research group for language LOTOS, and studying its practicability.

We study LOTOS itself, write a real OSI protocol in LOTOS, propose converting method from ASN.1 to LOTOS's ADT, and design LOTOS system.

In this paper, we present Intermediate Notation for LOTOS used by our system. This Notation is designed as general as any kind of "LOTOS system" can use it.

This Notation contains syntax and static semantics information of analyzed LOTOS text.

1 はじめに

LOTOSは、形式記述技法(FDT)の一つであり、OSI(解放型システム間相互接続)の仕様を厳密に記述する目的で、既に国際規格となっている[1]。

我々は、主にLOTOSの実用性に関する調査/研究を目的として、INTAPの下で研究会を行ってきた。その活動内容は、LOTOSの勉強[7]、実際のプロトコルとしてACSEのLOTOSによる記述実験[8]、ASN.1[2]からLOTOSの抽象データ型への変換の検討[9]、"LOTOSの処理系"の検討及び設計[10]などである。

本稿では、我々の設計しているLOTOS処理系の扱う中間言語の概要を紹介する。

2 形式記述技法(言語)とは

形式記述技法(FDT)は、通信プロトコルの国際規約における、厳密な仕様記述の必要から研究され発展した言語である。

2.1 自然言語による仕様記述(とその問題点)

ここでは、自然言語による仕様記述

100円を入れるとジュースかコーヒーを買うことができる自動販売機 ……(*)

を考える。

これを図1のように表わす。

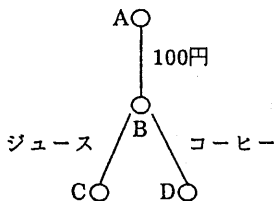


図1 自動販売機の問題点

図1は、Aが初期状態を表わし、システム(自動販売機)が100円を受取り、その内部状態はBに移り、状態Bからジュースを出すことにより、状態Cに移り、コーヒーを出すことにより、状態Dに移るということを表わしている。

一般に、あるシステムを外部から観測したときの動作を図1のような、非決定性有限状態オートマトンで表わしたとき、これをラベル付き状態遷移システム(LTS)と呼ぶ。

2.2 LOTOSによる仕様記述

LOTOSはLTSを記述するための言語であり、先の図1は、例えば、

```
process DrinkMachine [ in_100_yen, out_juice,
out_coffee ] :=
    in_100_yen; ( out_juice; stop
                []
                out_coffee; stop )
endproc
```

などとなる。

3 自然言語の問題点

3.1 従来のソフトウェア開発とその問題点

(1) 仕様が曖昧である。

自然言語による仕様記述(*)は一見完全の様に見えるが、実は、以下のシステムも仕様(*)を満足している。

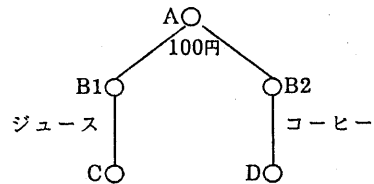


図2 自動販売機の問題点その2

図2のような自動販売機、すなわち100円を入れた段階で何らかの選択が行われ、システムが勝手に(例えば交互に)ジュース又はコーヒーを出すようなもの、も考えられる。

(2) テストに人手を要する。

従来のテストケースの生成では、人間が仕様(*)を読んで、

テストケース

- ・100円を入れてジュースを買うことができたか?
 - ・100円を入れてコーヒーを買うことができたか?
- を生成し、テストを行っていた。

(3) LOTOSの目指すもの

上記で述べたようなことを解決する。すなわち、仕様から曖昧性を排除し、機械的方法でのテストケースの生成、システムの正当性の数学的方法での検証、などである。

4 LOTOS処理系

4.1 処理系に求められるもの

処理系に求められる要求事項は、以下の二つの命題に集約される。

(1) 如何にして正しいLOTOS記述を得るか?

この要件を満足するためのツールとしては、構文エディタ、プリティプリンタ、デバッガ、(シミュレータ、インタプリタ)などがあげられる。

(2) 正しいLOTOS記述を如何にして使うか?

この要件を満足するためのツールとしては、コンパイラ、クロスリファレンスジェネレータ、テストケースジェネレータ、(シミュレータ、インタプリタ)などがあげられる。

4.2 既存の処理系

既存の処理系には、[3]、[4]、[5]などが知られている。これらは基本的には、LOTOSのサブセットを対象としたシミュレータであり、その操作は対話型が中心である。

4.3 我々の処理系

正直なところ、LOTOSの処理系とは何を指すかについては、研究会内でも必ずしも合意が得られていない。しかし、何でも(シミュレータ、ジェネレータ…)対応ができるものとしたい。

さて、LOTOSのどのような処理系を作るにしても、そのテキストファイルを読み込んで、字句/構文/意味を解析し(エラーを出し)、コンピュータ上で処理しやすい形の中間形式をもつことは必要である。

この中間形式、及びその操作パッケージの集合を汎用的に決めることで何にでも対応できるような処理系(LOTOSサーバ)を作り、LOTOS中間言語の設計を目標とする。

5. 中間語概要

この中間語は一口で言えば、
LOTOS構文解析木 + 静的意味
を表現するものである。

以降の説明では、便宜上、中間語をノードとし、中間語同志の論理的な関係を、ノード間を結ぶ有向エッジとして表現する。但し、これらは計算機上で

のポインタによる表現を仮定するものではない(計算機上での実現方法はここでは特に規定しない)。

付図1はLOTOSのテキストから生成された中間語の様子を表す概念図である。

5.1 中間語

中間語(中間言語)とは、入力テキストを解析することにより生成される内部的な表現であり、標準的なものとしては、AdaにおけるDIANAが有名である。[6]

この中間語は、LOTOSサーバによって生成される木構造のデータ構造であり、LOTOSを利用するアプリケーションが必要とする各種の情報を保持するためのものである。これらの情報は、サーバがLOTOSテキストを解析することによって得られる。尚、この中間語については、以下のことが言える。

- (a) 本中間語は、入力されたLOTOSテキストに比例した情報量を持つ。
- (b) 本中間語は、入力されたLOTOSテキストの持つ、意味のある情報を、すべて持つ(冗長な括弧付けや空白類文字、コメントの内容などの情報は捨てられる)。
- (c) 本中間語は、シミュレーションやテストケースの生成などの際に必要となる付加情報を持つ。

クライアントは、LOTOS仕様のシミュレーションやLOTOS仕様からのテストケースの生成などを行う場合、サーバ側から、中間語の操作を行うサービス(関数)が提供されるので、これらを用いて各種の処理を行うことができる。尚、中間語に関する操作とは、以下の様なものを言う。

- (d) 中間語から情報を取り出す。
- (e) 中間語の形を判定する。
- (f) 中間語に情報を設定する。
- (g) 中間語を変換する。

中間語はその性格上、次の3つに分類される。

- (1) 構造系中間語
- (2) ADT中間語
- (3) CCS中間語

5.1.1 構造系中間語

構造系中間語とは、LOTOS仕様の構造を反映するための中間語であり以下の種類がある。(付図2)

- specificationノード
- behaviourノード
- スコープノード
- 補助ノード

ただし、補助ノードは計算機上でのインプリメントを意識したものであって、本質的にはbehaviourノード及びスコープノードの一部である。

5.1.2 ADT中間語

ADT中間語とは、LOTOS仕様のうち抽象データ型の定義部の情報を保持するための中間語であり、以下の種類がある。

- typeノード
- sortsノード
- operationノード
- equationノード

付図3は、ADT部に関する構文/意味解析後の中間語の様子である。

5.1.3 CCS中間語

CCS中間語とは、LOTOS仕様のうち動作定義部の情報を保持するための中間語であり、以下の種類がある。

- 動作式ノード(各種)
- processノード
- gateノード
- 仮引数ノード

付図4及び付図5は、CCS部に関する構文/意味解析後の中間語の様子である。

6. 今後の課題

まず何よりも、この処理系を完成させることが重要である。中間語の汎用性については、かなりよい近似解であると信じている。とは言え、真の意味での汎用性を持たせるには、まだ問題も多く残されている。今後は、この中間語をベースとして、真の意味での汎用中間語すなわち、標準中間語に発展させていきたい。

7. 謝辞

LOTOS研究会は多くの方々の御協力により開催された。各社の関係者の方々には、LOTOS研究会に快くメンバを出して頂いた。(財)情報処理相互運

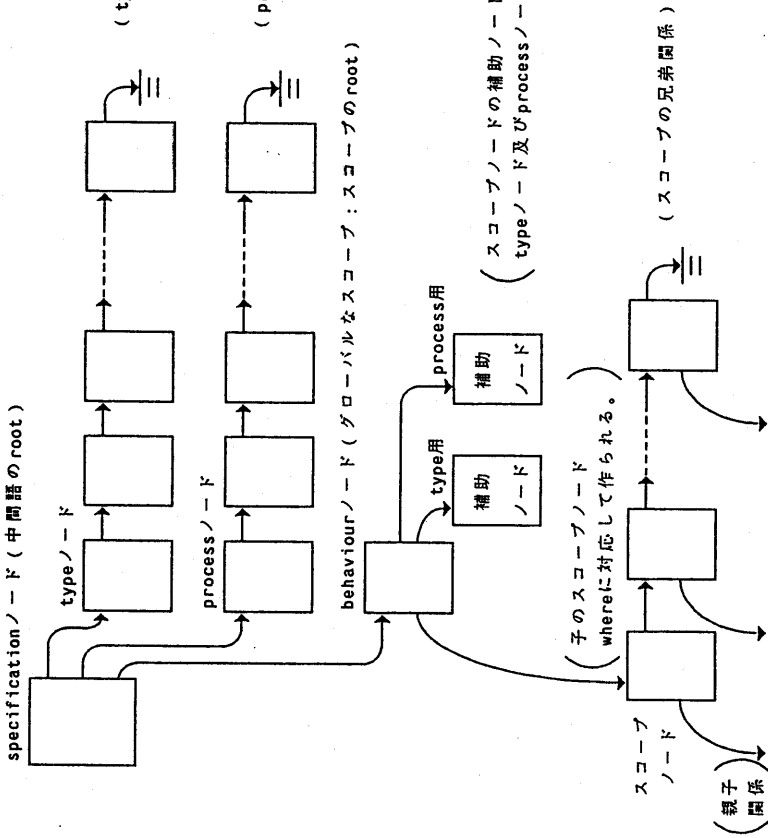
用技術協会には、研究会の開催にあたり、各種の便宜を図って頂いた。

ここに記して感謝の意を表したい。

8. 参考文献

- [1] ISO 8807: OSI-LOTOS-A formal description technique based on the temporal ordering of observational behaviour
- [2] ISO 8824: OSI - Specification of Abstract Syntax Notation one(ASN.1)
- [3] J.P.Briand, M.C.Fehri, L.Logrippo, A.Obaid: EXECUTING LOTOS SPECIFICATIONS, Protocol Specification, Testing, and Verification, VI, pp73-84(1987)
- [4] R.Guillemot, M.Haj- Hussein and L.Logrippo: Executing Large LOTOS Specifications, Protocol Specification, Testing and Verification 8, pp 339-410(1988)
- [5] ESPRIT/SEDOS: HIPPO-LOTOS simulator, ESPRIT/SEDOS-ST 410
- [6] 細谷監修: Adaコンパイラ実現のための中間言語DIANA入門/言語仕様/応用,啓学出版,1986
- [7] 大薮、他: "哲学者の食事問題のLOTOSによる記述実験",情報処理学会ソフトウェア工学研究会 66-4,(1989)
- [8] 内山、藤田、他: "ACSEのLOTOSによる記述の試み",情報処理学会マルチメディア通信と分散処理研究会,(1990) 発表予定
- [9] 五ノ井、他: "ASN.1からLOTOS ADTへの変換方法",情報処理学会マルチメディア通信と分散処理研究会,(1990) 発表予定
- [10] 辻、佐藤、他: "クライアント/サーバモデルに基づくLOTOS仕様記述支援システム的设计",情報処理学会マルチメディア通信と分散処理研究会,(1990) 発表予定

構文/意味解析後の中間語の様子

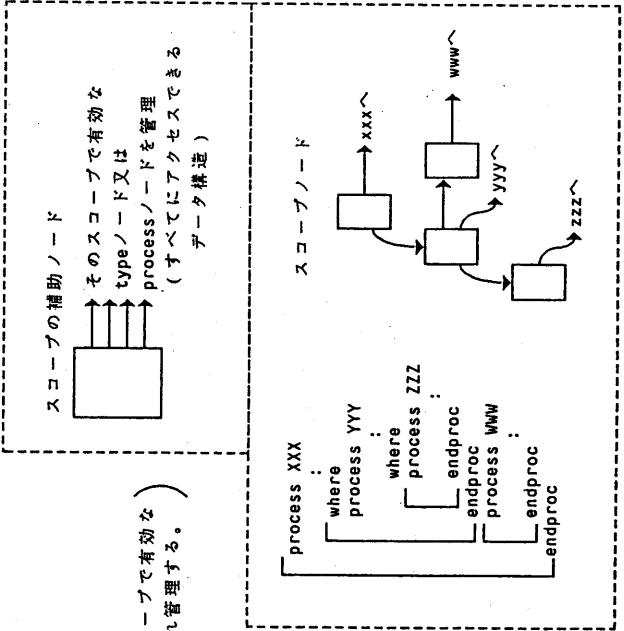


(typeノードの線形リスト: 全てのtypeノードを宣言順に管理)

(processノードの線形リスト: 全てのprocessノードを宣言順に管理)

behaviourノード (グローバルなスコープ: スコープのroot)

(スコープノードの補助ノードはそのスコープで有効なtypeノード又はprocessノードを管理(すべてにアクセスできるデータ構造))



付図2

(1) specification ノード

| |
|------------|
| Imitag |
| Attribute |
| Parameters |
| Name |
| Lineno |
| Filename |
| Types |
| Processes |
| Behaviour |

(3) スコープノード

| |
|---------------|
| Imitag |
| Attribute |
| Types-Aux |
| Processes-Aux |
| Lineno |
| Filename |
| Parentnode |
| Leftnode |
| Rightnode |

(2) behaviour ノード

| |
|---------------|
| Imitag |
| Attribute |
| Types-Aux |
| Processes-Aux |
| Lineno |
| Filename |
| Parentnode |
| Leftnode |
| Rightnode |

(4) 補助ノード

| |
|---------|
| Imitag |
| Members |
| Next |
| Element |
| Element |
| : |
| : |
| : |

付図3

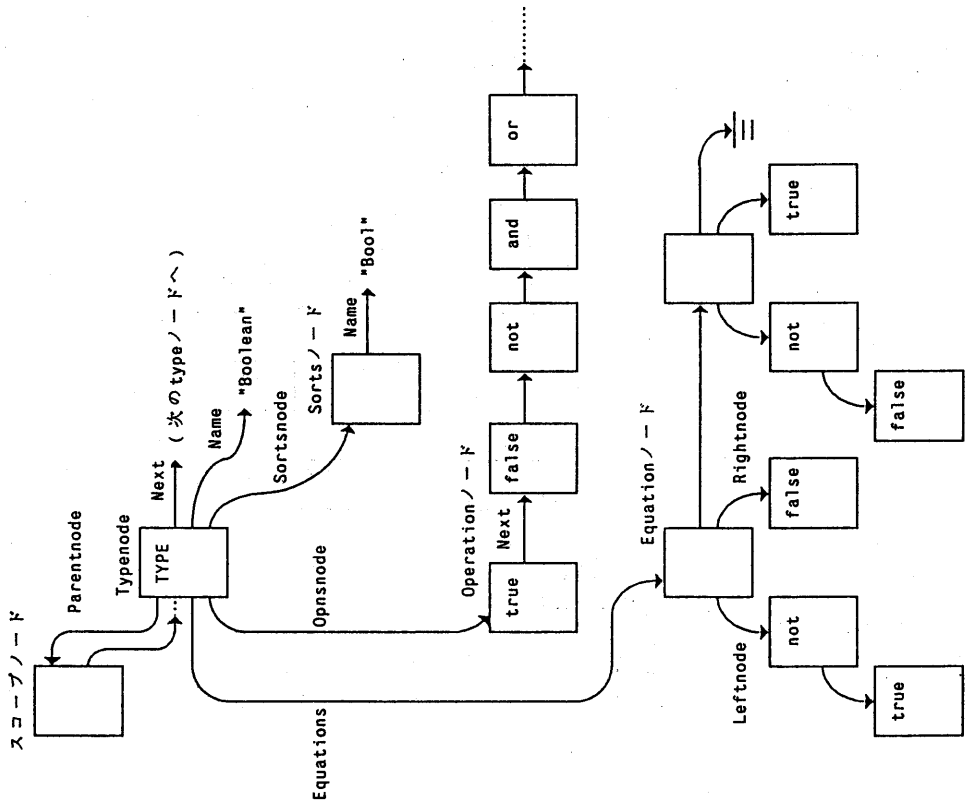
【例】 (ISO8807-AnnexA P72)

```

type Boolean is
  sorts Bool
  opns true, false
  not
  _and_, _or_, _xor_,
  .....
  : Bool, Bool -> Bool

eqns forall x, y : Bool
  ofsort Bool
  not(true) = false;
  not(false) = true;
  x and true = x;
  x and false = false;
  x or true = true;
  x or false = x;
  :
  :
  :
endtype

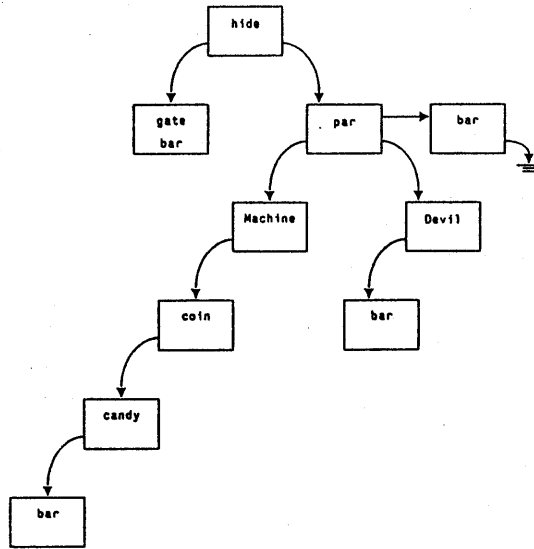
```



付図4

```

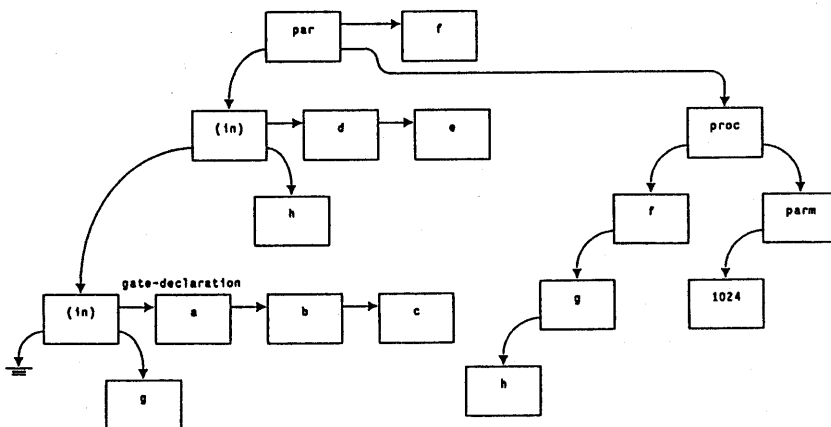
【例】
hide bar in
Machine [ coin, candy, bar ]
! [ bar ]
Devil [ bar ] .
    
```



付図5

```

【例】
par g in [ a, b, c ], h in [ d, e ] ! [ f ] proc [ f, g, h ] ( 1024 )
    
```



(注) 本ノードに簡略化を適用することで
(7)parallel-expressionノードに展開される。