

OSI 通信ソフトウェアにおける 試験データの自動生成

辻 宏郷 佐藤 文明 勝山 光太郎 水野 忠則

三菱電機 (株) 情報電子研究所

我々は OSI 通信ソフトウェアの開発を支援するために、SDL と ASN.1 を組み合わせた仕様記述を用いた 適合性試験システムを作成してきた。FDT 適用目的の一つは 設計仕様から試験仕様の自動生成 であるが、試験系列に対応する試験データの作成 は 試験実行者が行う必要があり、また テスタの試験対象依存部 の開発コストが 問題となっていた。本論文では、SDL と ASN.1 を補足する パラメータ仕様記述言語 PSL.1 を提案し、これらの FDT を組み合わせた記述を用いて 試験データを自動生成する ASN.1 試験データ生成 / 解析ツール を開発した。また、応用層プロトコルの一つである OSI トランザクション処理 の 試験への適用方法を示す。

Automatic Test Data Generation of OSI Communication Software

Hirosato TSUJI Fumiaki SATO Kotaro KATSUYAMA Tadanori MIZUNO

Computer & Information Systems Laboratory
MITSUBISHI ELECTRIC CORPORATION

5-1-1 Ofuna , Kamakura-City , Kanagawa 247 , JAPAN

We have been developing the communication software testing environment FOREST, based on the OSI Conformance Testing Methodology. Currently, the test executors must generate the test data for the test sequences which are generated automatically from SDL specification. On the other hand, the implementation cost of tester modules is increasing. In this paper, we propose the Parameter Specification Language PSL.1 for an extension of ASN.1. And we have developed ASN.1 Test Data Generation Tools, which realize automatic test data generation from the specification using such FDTs. Furthermore, we present their application methodology to the OSI Transaction Processing software testing.

1 はじめに

ISO（国際標準化機構）における OSI（開放型システム間相互接続）の標準化が進展しており、各地で OSI 通信ソフトウェアの実装が試みられている。実装したソフトウェアに対する OSI 規格への適合性試験は不可欠であり、試験コストや試験期間の削減が重要となっている。一方、OSI 通信プロトコルやサービスの仕様を曖昧なく定義するための FDT（形式記述技法）として、ISO によって Estelle および LOTOS、CCITT（国際電信電話諮問委員会）によって SDL [1] の開発が行われている。また、OSI 応用層やプレゼンテーション層におけるデータ（PDU：プロトコルデータ単位）を定義するための言語として、抽象構文記法 ASN.1 [2][3] が、ISO で開発されている。

我々は、OSI 通信ソフトウェアの開発を支援するために、SDL と ASN.1 のサブセットを組み合わせた形式的仕様記述 [11] を用いた、OSI 適合性試験法 [4]（図 1）に基づく試験システム FOREST [7] を開発してきた。SDL は状態遷移モデルに基づく FDT であり、開発対象ソフトウェアの規格書に記述された状態遷移表と相性がよい。また、データ構造は ASN.1 を用いて定義されているため、規格書の記述をそのまま引用することが可能である。仕様記述に FDT を用いる目的は、1) 仕様を曖昧なく定義することによる設計者と実装者が一意に解釈可能な仕様の獲得、2) 形式仕様からプログラムの自動生成 [10]、3) 設計段階の要求仕様から試験仕様（試験系列、試験データ）の自動生成、等がある（図 2）[6]。

FOREST においては、試験系列の自動生成が実現されているが、試験データの作成は、ASN.1 ツール APRICOT [9] のデバッガやパラメータ設定ツールを用いて、試験実行者が対話的に行う必要があった。また、試験対象のプロトコルが複雑になるに伴い、テスト内部に特別な処理を行なうモジュールを準備する必要性が生じ、その開発コストが問題となっている。

本研究では、これらの問題点を解決するために、SDL + ASN.1 と組み合わせて仕様記述を行う形式的仕様記述言語を提案し、また試験データの自動生成を実現するツールを開発した。以下、2 章では OSI 通信ソフトウェアに対して試験を実施する際の試験データに関連する問題点を述べる。3 章では、これらの問題点を解決するための仕様記述言語 PSL.1 を提案する。また 4 章では、ASN.1 試験データ生成／解析ツールについて述べ、5 章で応用層プロトコルの一つであるトランザクション処理の試験への適用方法を示す。

2 試験実施上の問題点

2.1 試験データの作成に関する問題

OSI 通信ソフトウェアに対する試験仕様は、試験系列と試験データから構成されている。

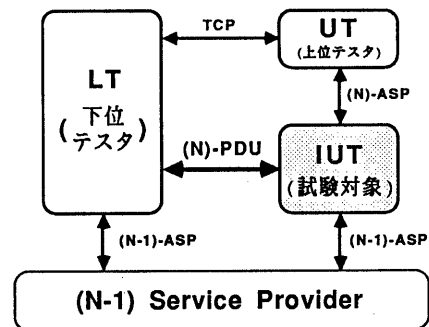


図1 OSI 適合性試験法

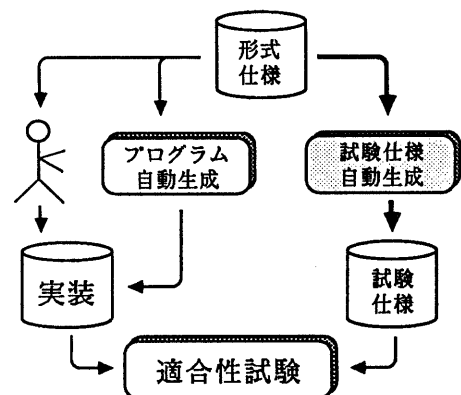


図2 形式仕様からの実装と試験

試験系列は 試験対象 (IUT) と テスタ (UT, LT) 間で交換するイベントの順序関係であり、試験データはイベントにおいて交換すべき PDU である。形式仕様から試験仕様の作成過程は、二つの段階に分けることができる (図3)。第一の段階は、形式仕様から試験系列の生成である。FOREST においては、状態遷移モデルに基づく形式仕様 (SDL) から TTCN [4] で記述された試験系列を自動生成する [8]。第二の段階は、試験系列に対応する試験データの作成である。ASN.1 を用いて定義された PDU 構造 (図4) に対して、試験系列を実行するために必要なパラメータを設定することによって、試験データを作成する。

現在は、このパラメータをユーザが対話的に設定する必要がある、試験データ作成は自動化されていない。OSI 応用層プロトコルにおいては、PDU に含まれるパラメータ値が重要な意味を持っている。例えば、異なった種類の命令を同一の PDU で表現している場合があり、そのパラメータ値によって意味を区別している場合がある。このため、自動生成された試験系列を利用するためには、試験に必要なパラメータ値を的確に設定した試験データの作成が必須となっている。

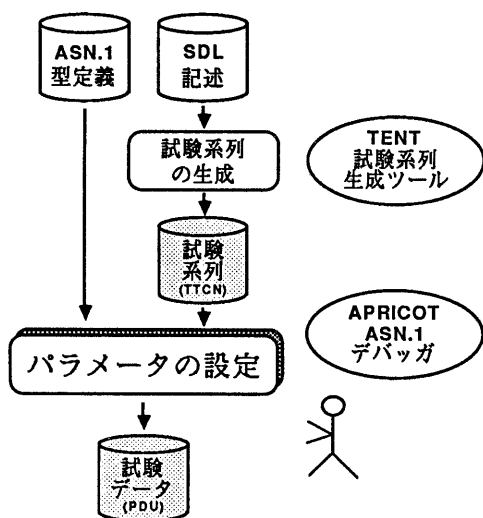


図3 試験データの生成過程

このパラメータ設定に関する情報は、規格書においては自然言語等を用いて非形式的に記述されており、計算機処理が可能な形で表現されていない。従って、試験データ作成時のパラメータ設定を自動化するためには、パラメータ設定条件を計算機の理解できる形で形式的に記述する必要がある。試験データの仕様を形式的に記述する方法は、ASN.1 の値記法を用いてパラメータ値を記述することである。一般に試験データに含まれるパラメータは、

- 1) 特定の値を取るべきパラメータ
- 2) ある値集合の中からいずれかを選択すべきパラメータ
- 3) どのような値を取っても試験に影響の無いパラメータ

に分類することができる。しかし、ASN.1 値記法を用いてパラメータを記述すると、全てのパラメータに対する設定条件を記述しなければならない (図5)。このため、分類 2) や 3) に属するパラメータ値に対する記述も必要となり、試験実行準備の作業量が增大する。また試験において本当に検査すべき項目と、検査対象でない項目とが曖昧になりやすい。

```

TP-BEGIN-DIALOGUE-RI ::= SEQUENCE {
  CHOICE {
    channel [1] SEQUENCE {
      channel-identifier [1] Channel-id ,
      channel-utilization [2] ENUMERATED
        { tppm-recovery (1)
        } DEFAULT tppm-recovery
    } ,
    dialogue [2] SEQUENCE {
      dialogue-identifier [1] Dialogue-id ,
      recipient-tpsu-title [2] TPSU-TITLE ,
      initiating-tpsu-title [3] TPSU-TITLE
        OPTIONAL ,
      initial-coordination-level
        [4] ENUMERATED
        { none (0) ,
          commit (1)
        } DEFAULT commit ,
      selected-FUs [5] FU-List
        DEFAULT {} ,
      invocation-data User-inf OPTIONAL
    }
  }
}
  
```

図4 ASN.1 型記法を用いた PDU 定義例

2.2 試験データの運用に関する問題

OSI 通信ソフトウェアに対する試験において必要となる試験データは、二種類に分類することができる。一方は LT (下位テスト) が IUT (試験対象) に対して発行する PDU であり、他方は LT が IUT から受信することが期待される PDU である (図 6)。前節で述べた様に、試験において検査すべき項目は試験データ全体とは限らず、特定のパラメータだけを検査すればよい場合がある。またそのパラメータの値は、一定値とは限らずに集合内の値、任意の値の場合もあり得る。従って、LT が IUT から受信するデータについては、あらかじめ期待される試験データを作成しておき、受信データとの単純な一致検査を行うことができない。

また、応用層の複雑なプロトコルによっては、UT (上位テスト) や LT から制御不可能な IUT の振るまいによって、LT がその振るまいに応じた試験データやそのパラメータを必要とする場合がある。例えば、OSI トランザクション処理 [5] の試験において、IUT からダイアログを確立する試験シナリオの場合、ダイアログ ID は IUT のプロトコルマシン内部で決定し、外部から制御不可能である。このダイアログ ID の値は、LT から IUT に発行する試験データにおいても用いなければならない。従って、LT から IUT に発行するデータについても、あらかじめ全ての試験データを作成しておくことは不可能である。

このような複雑な試験シナリオに対応するためには、試験対象が変わる毎にテスト (UT, L

```
{ tp-begin-dialogue-ri {
  dialogue {
    dialogue-identifier      0      ,
    recipient-tpsu-title     3      ,
    initiating-tpsu-title    2      ,
    initial-coordination-level none ,
    selected-FUs              '0110'B
  }
}
```

図 5 ASN.1 値記法を用いた PDU 記述例

T) 内部に特別な処理を行なうモジュールを組み込む必要があり、テスト開発に要する作業量を増大させる原因となっている。

2.3 試験実施のための解決策

前節で述べた問題点を解決するために、我々は、試験に必要な設定条件のみを形式的に記述するための簡易言語を定義し、これを用いてパラメータ設定条件を記述することとした。この言語は、実装時の効率を重視して APRICOT の ASN.1 プリコンパイラで削除した定義情報を補足する意味をもつ。また、この記述を利用して試験データを自動生成し、様々な試験形態に柔軟に対応するツールを開発する。

3 PSL.1 の提案

3.1 パラメータ設定時の非決定性要因

ASN.1 を用いてデータ構造を定義された PDU に対して、パラメータを設定する際に人手の介入を要する非決定性は、以下の四通りに分類することができる。

- 1) ある型のパラメータ値
- 2) 省略可能 (OPTIONAL) な型に対するパラメータの設定/省略
- 3) 単一型の順列/集合で表現される型 (SequenceOf 型, SetOf 型) の要素の個数
- 4) 複数の要素 (Choice 型) から一つの要素の選択

1) の非決定性に対する回答は、(a) 特定の値を取るべき (b) ある値集合に含まれるいずれかの値を取るべき (c) どのような値を取ってもよい、

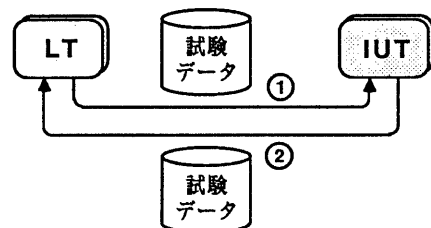


図 6 LT, IUT と試験データ

のいずれかである。2)の非決定性については、(a) 設定しなければならない (b) 省略しなければならない (c) どちらでもよい、のいずれかである。また、3)の非決定性については、正の整数値についての1)に等しく、4)の非決定性については、有限値の集合 (Enumerated 型) の値についての1)に等しい。

3.2 PSL.1

PSL.1 (Parameter Specification Language for ASN.1) は、試験系列 から試験データを作成する際の ASN.1 パラメータ設定において発生する、全ての非決定性に対する回答を記述するための言語である。PDU の各パラメータの識別は、APRICOT のパス形式を用いて指定する。

例えば、type 型であるパス path のパラメータとして value の値を設定すべき場合、

```
type ( path ) = value ;
```

と記述する。この構文を用いて、前節 1) の非決定性に対して、(a) に属するパラメータについて設定条件を記述する。PSL.1 では、等号の代わりに不等号を用いることができる。従って、

```
enum(diagnostic) >= 1 ;
```

```
enum(diagnostic) <= 3 ;
```

の様に 複数の条件式の積によって、(b) に属するパラメータの設定条件を記述する。

2) の非決定性に対してパラメータを省略することを明示的に指定する場合は、

```
bit(selected-FUs) = OPT ;
```

と記述する。3) の要素数の指定については、

```
seqof(user-data) = 3 ;
```

と記述する。4) の要素の選択については、

```
choice(tp-bgn-dlg) ;
```

と記述する。

また、PrintableString 型等の文字列型や、OctetString 型等のオクテット列で表現されるパラメータに対しては、設定すべきパラメータを別のファイルに作成しておき、

```
printable(text) < "doc" ;
```

の様に参照することが可能となっている。

4 ASN.1 試験データ生成/解析ツール

4.1 ツール構成

我々は、前章で提案した PSL.1 を用いたパラメータ設定条件記述を利用するツールとして、ASN.1 試験データ生成/解析ツールを開発した。このツールは、試験データ生成ツール *tdgen* と試験データ解析ツール *tdana* の二つのプログラムから構成されている (図 7)。各プログラムは C 言語を用いて記述しており、ASN.1 の符号化/復号化処理については、APRICOT のコーデックライブラリを利用している。

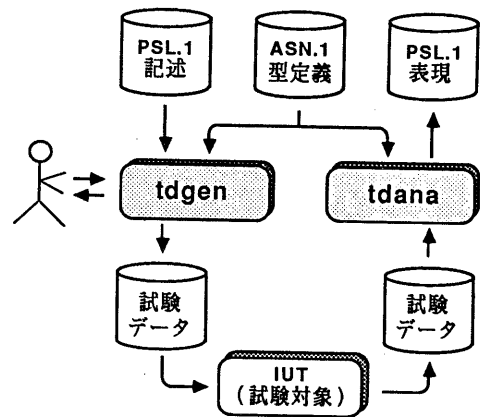


図7 試験データ生成/解析ツール

4.2 試験データ生成ツール *tdgen*

試験データ生成ツール *tdgen* は、PDU の構造を定義した ASN.1 データ型定義テーブル (ASN.1 プリコンパイラの出力) と、パラメータ設定条件を記述した PSL.1 記述を入力とし、不足する情報をユーザと対話的にあるいは自動的に補足することによってパラメータを決定し、符号化した試験データ (PDU) を作成するツールである。設定パラメータは、

(1) PSL.1 で記述した設定条件

(2) ユーザとの対話的入力

(3) 非決定解決アルゴリズム

の優先順位に従って決定する。すなわち、PSL.1 を用いてパラメータ設定条件等 (図 8) が記述されている場合は、その指定に従う。記述が

存在しない場合、対話モードではユーザに非決定性の解決（値の入力、選択等）を要求し、解決拒否（改行キーのみ入力）ならば、後述の非決定性解決アルゴリズムに従って、パラメータを自動決定する（図9）。全自動モードでは非決定性に対するユーザとの対話入力は省略する。すなわち、PSL.1記述の存在しない全ての非決定性に対して非決定性解決アルゴリズムを適用することによって、試験データの自動生成を行う（図10）。

また **tdgen** は、ユーザとの対話あるいは自動解決によって得られた非決定性に対する回答を、PSL.1の構文形式で出力する機能を持っている。このPSL.1表現をOSのリダイレクション機能を用いてファイルに保存すること

```
choice(ri-bgn-dlg) ;
choice(ri-bgn-dlg/elem/dialogue) ;
int(ri-bgn-dlg/elem/dialogue/dlg-id) = 0 ;
int(ri-bgn-dlg/elem/dialogue/rcp-tpsu-ttl) = 3 ;
int(ri-bgn-dlg/elem/dialogue/int-tpsu-ttl) = 2 ;
enum(ri-bgn-dlg/elem/dialogue/init-co-lvl) = 0 ;
bit(ri-bgn-dlg/elem/dialogue/slcted-FUs) = "0110" ;
seqof(ri-bgn-dlg/elem/dialogue/inv-data) = OPT ;
```

図8 OSI-TP のPDU用 PSL.1 記述例

```
% tdgen -t > TP-BGN-DLG.exe
=====
ASN.1 Test Data Generator (1990/11)
=====
Input ASN.1 Table File Name ? > TP.t
Input ASN.1 Type/Path to Encode ? >
Input Extended Spec File Name ? >

[CHOICE] There are 22 choices at "...
  1) ri-bgn-dlg
  2) rc-bgn-dlg
  :
  22) rc-bd
Please Input Node Number ? (1-22) > 1

[CHOICE] There are 2 choices at "ri-bgn-dlg/elem".
  1) channel
  2) dialogue
Please Input Node Number ? (1-2) > 2

[INT] ri-bgn-dlg/elem/dialogue/dlg-id ? > 0

[INT] ri-bgn-dlg/elem/dialogue/rcp-tpsu-ttl ? > 3

[INT] ri-bgn-dlg/elem/dialogue/int-tpsu-ttl [OPT] ? > 2

[ENUM] ri-bgn-dlg/elem/dialogue/init-co-lvl ? > 0

[BIT] ri-bgn-dlg/elem/dialogue/slcted-FUs [OPT] ? > "0110"

[SEQOF] Path "ri-bgn-dlg/elem/dialogue/inv-data" is OPTIONAL.
Do you want to make This Node ? (y/n) > n
```

図9 OSI-TP のPDU 対話生成例

によって、エディタを使用せずに簡単に PSL.1 記述を作成することができる。さらに、ASN.1 定義テーブルや PSL.1 記述の解析だけを実行する機能等を持っている。

パラメータを自動決定する場合は、同値分割法、境界値解析法 [12] を応用した非決定性解決アルゴリズムを用いて決定する。時刻等を表現するパラメータに対しては、妥当な値を選択するための特別なモジュールを実装している。

4.3 試験データ解析ツール **tdana**

試験データ解析ツール **tdana** は、ASN.1 データ型定義テーブルと試験データ (PDU) を入力とし、復号化を行いパラメータ解析情報を出力するツールである。この時、解析情報 (パラメータ値等) を PSL.1 構文の書式で出力する機能をもっており、この PSL.1 表現を **tdgen** の入力ファイルとして再利用することができる。**tdgen** と **tdana** を組み合わせることによって、試験対象の動作に応じた試験データの作成や、検査必要項目のみの試験の実行が可能となる。

```
% tdgen -th TP.t TP-BGN-DLG TP-BGN-DLG.exe
=====
ASN.1 Test Data Generator (1990/11)
=====
ASN.1 Definition File Name : "TP.t"
ASN.1 Type to encode Test Data : TPASE-APDU
ASN.1 Path to encode Test Data :
Test Data File Name to Generate : "TP-BGN-DLG"
Extended Specification File : "TP-BGN-DLG.exe"
Generation Mode No. : 0040

... Extended Specification ...
[CHOICE] "ri-bgn-dlg" at ""
[CHOICE] "dialogue" at "ri-bgn-dlg/elem"
[SEQOF/SETOF] "ri-bgn-dlg/elem/dialogue/inv-data" is OPT
[INT/ENUM/BOOL] "ri-bgn-dlg/elem/dialogue/dlg-id" = 0
[INT/ENUM/BOOL] "ri-bgn-dlg/elem/dialogue/rcp-tpsu-ttl" = 3
[INT/ENUM/BOOL] "ri-bgn-dlg/elem/dialogue/int-tpsu-ttl" = 2
[INT/ENUM/BOOL] "ri-bgn-dlg/elem/dialogue/init-co-lvl" = 0
[BIT] "ri-bgn-dlg/element/dialogue/slcted-FUs" = "0110"

-----
ri-bgn-dlg/elem/dialogue/dlg-id = 0 (INT)
ri-bgn-dlg/elem/dialogue/rcp-tpsu-ttl = 3 (INT)
ri-bgn-dlg/elem/dialogue/int-tpsu-ttl = 2 (INT)
ri-bgn-dlg/elem/dialogue/init-co-lvl = 0 (ENUM)
ri-bgn-dlg/elem/dialogue/slcted-FUs = 0110 (BIT)

a0 12 a2 10 81 01 00 82 01 03 83 01 02 84 01 00
85 02 04 60
```

図10 OSI-TP のPDU 自動生成例

5 OSI-TP の試験への適用法

5.1 試験適用の準備

本章では、OSI トランザクション処理 (OSI-TP) に対する 試験実行時の問題点 について、現在の解決方法と **tdgen** / **tdana** 適用後の解決方法を示す。動作部は SDL サブセットを用いて記述し、PDU は ASN.1 プリコンパイラの構文を満たすように規格書の定義を修正し、プリコンパイラを用いて 型定義テーブル に変換しておく。動作部から得られる試験系列 (TTCN) に対して、従来は 表形式 等で記述していた設定パラメータ情報を PSL.1 で記述する。

5.2 試験データの即時生成

2.2 節で述べた様に、IUT からダイアログを確立した場合、LT は IUT から受信した TP-BEGIN-DIALOGUE-RI に含まれる ダイアログ ID を保存し、LT から IUT へ発行する TP-BEGIN-DIALOGUE-RC に同一の値を設定する必要がある。現在の試験では、ダイアログ ID の値に 0 を設定した試験データを作成しておき、受信したダイアログ ID の値を符号化箇所に直接設定するモジュールを、LT 内部に実装することで対処している。

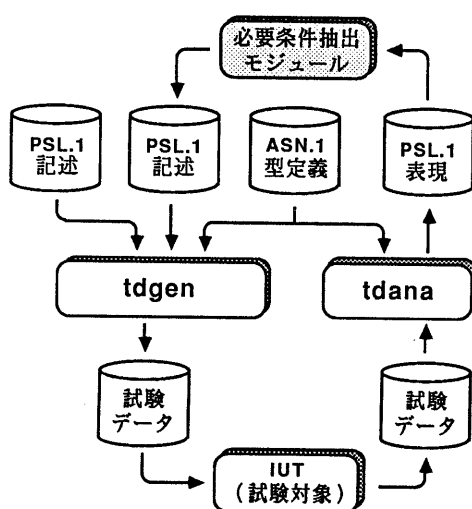


図 11 試験データの即時生成

この方式では、解析情報の再利用項目が増加する度にテスト内部に専用処理モジュールを作成する必要が生じ、テストの汎用性が低下する。そこで、PSL.1 構文で表現された試験データ解析情報から試験データ作成に必要な式を取り出す 必要条件抽出モジュール を作成する。**tdgen** では PSL.1 を複数のファイルに分割して記述することが可能であり、IUT の動作によって変化しないパラメータ設定条件は、あらかじめ PSL.1 記述で作成しておく。そして、受信データを PSL.1 表現に逆変換して抽出した式の一部と、用意しておいた PSL.1 記述を組み合わせることで **tdgen** の入力 とすることによって、試験対象の振るまいに応じて試験データの即時作成を行なう (図 11)。

5.3 PSL.1 変換比較

IUT から LT が受信することが期待される PDU について、全てのパラメータが検査対象項目とは限らない。例えば、IUT 側からのダイアログ確立時の診断パラメータは、テスト側では予測することができない。理想的には受信 PDU を復号化して必要項目のみ検査することが望ましいが、このためには 全ての受信データに対する 専用の検査モジュールが LT 内に必要となる。現在の試験では、規約上不定 (実装者依存) のパ

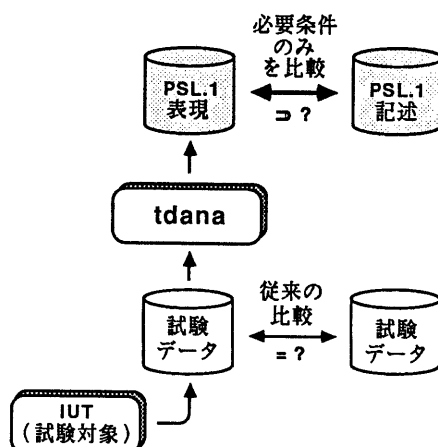


図 12 試験データの PSL.1 変換比較

ラメータ値については、プログラムを作成した開発者から内部動作に関する情報を入手し、全てのパラメータの期待値を明らかにした上で、単純に一致検査すればよい試験データを用意する等の対策を行っている。

このような試験シナリオを作成すると、試験において検査すべき項目と任意の値で構わない値の区別が曖昧となる。そこで、試験データを必要な検査項目のみ記述した PSL.1 構文形式で用意しておく。受信した PDU が用意しておいた PDU と一致することを検査する代わりに、PSL.1 表現に逆変換して PSL.1 構文の式ごとに比較し、試験データの必要条件全てが PDU に含まれていることを検査する (図 12)。

6 おわりに

SDL と ASN.1 の組み合わせた形式仕様記述において、パラメータ設定条件を記述するための PSL.1 を提案した。これらの FDT を組み合わせた記述を用いて試験データを自動生成する ASN.1 試験データ生成/解析ツールを開発した。PSL.1 を用いることによって、試験実行に必要な仕様情報を簡潔かつ厳密に記述できるようになった。tdgen / tdana を試験データ作成・解析に用いることによって、試験対象ごとにテストに用意すべきロジックが減少し、テストの汎用性を高めて開発コストの低減が可能となる。今後の課題は、自動生成する試験データの品質を向上させると共に、試験実行者が望む検査範囲の網羅性に対応することである。また、現在は試験仕様の生成のみに適用しているが、SDL を用いた状態遷移記述と PSL.1 を用いたパラメータ記述の融合を行うことによって、プログラム自動生成への適用を考察する計画である。

参考文献

[1] CCITT : Functional Specification and Description Language, Recommendation Z.100 (1989).
[2] ISO/IEC : OSI - Specification of Abstract Syntax Notation One (ASN.1), ISO 8824 (1990).

[3] ISO/IEC : OSI - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), ISO8825 (1990).
[4] ISO/IEC : OSI Conformance Testing Methodology and Framework, ISO/DIS9646 (1989).
[5] ISO/IEC : OSI - Distributed Transaction Processing, ISO/DIS10026 (1989).
[6] T.Mizuno, J.Munemori, F.Sato, T.Nakakawaji, K.Katsuyama : COTTAGE : Systematic Method for the Development of Communication Software, *Protocol Specification, Testing and Verification VIII, North-Holland*, (1988).
[7] K.Katsuyama, F.Sato, T.Nakakawaji, T.Mizuno : OSI Testing Environment Based on the Standardized Formalisms, *Third International Conference on Formal Description Techniques (FORTE'90)*, (1990).
[8] F.Sato, K.Katsuyama, T.Mizuno : TENT : Test Sequence Generation Tool for Communication Systems, *Second International Conference on Formal Description Techniques (FORTE'89)*, (1989).
[9] T.Nakakawaji, K.Katsuyama, N.Miyauchi, T.Mizuno : Development and Evaluation of APRI-COT (Tools for Abstract Syntax Notation One), *Proceedings of the second international symposium on interoperable information systems*, (1988).
[10] K.Tanaka, F.Sato, K.Katsuyama, T.Mizuno : AGAIN : Application protocol Generator to Accelerate Implementation Network Software, *Fifth SDL Forum*, (1991) 発表予定.
[11] 田中、辻、佐藤、水野 : SDL に基づく通信ソフトウェア開発環境, 情報処理学会第 41 回全国大会 4H-5, (1990).
[12] Glenford J.Myers : The Art of Software Testing, *John Wiley & Sons, Inc.*, (1979).
[13] 楠、田中、中川路、勝山、水野 : OSI トランザクション処理システムの試作, 情報処理学会マルチメディア通信と分散処理研究会 45-5, (1990).
[14] 長谷川、野村、堀内 : ASN.1 支援ツールの開発 - コンパイラおよびエディタ -, 情報処理学会マルチメディア通信と分散処理研究会 39-4, (1988).
[15] 長谷川、野村、瀧塚 : SDL と C を組み合わせた通信プログラム仕様の記述法 及びその処理系, 情報処理学会マルチメディア通信と分散処理研究会 46-5, (1990).