4L−8

パックト符号化規則対応のASN.1コンパイラの設計

長谷川 亨　野村 真吾
国際電信電話株式会社 研究所

1. はじめに

ASN.1(抽象構文記法1)[3]のデータ定義から基本符号化規則の符号化/復号プログラムを自動生成するASN.1コンパイラは、OSI応用プロトコル実装の効率化に寄与してきた[1, 5]。しかし、基本符号化規則で符号化されたデータは冗長なため、現在、CCITT, ISOはパックト、ライトウェイト等の新符号化規則の標準化を進めている。パックト符号化規則[4]は符号化の高速化に有効であるという報告もあり[2]、パックト符号化規則対応のコンパイラが望まれる。ただし、現在広く基本符号化規則が使用されているため、パックト符号化規則だけを提供するコンパイラは現実的でなく、双方の規則を同時に提供することが重要な条件である。筆者らは、基本、パックト双方の符号化/復号処理を1つのプログラムとして生成する方式を検討し、実現の見通しを得たので、本稿ではその概要を述べる。

2. アプローチ

2.1 基本およびパックト符号化規則

基本符号化規則(BER)は、全ての値を、識別子(ID)、長さ(LI)、コンテンツ(CO)の3つ組からなるオクテット列に符号化する。図1にSEQUENCE型の値から符号化されたオクテット列の構成を示す。構造型を用いて入れ子になる値は、親のCOに展開される。SEQUENCEの要素bには明示的なタグ[0]が付加されており、この明示的なタグはID,LIの組に符号化される。構造型のLIには、COの長さを示す固定長形式と、EOCをデリミタとして使用する不定長形式の2種類があり、例では不定長形式を用いている。

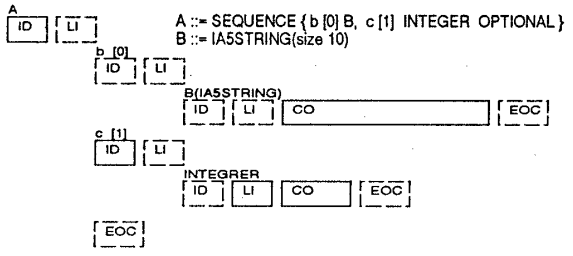

図1: 符号化オクッテト列の構成

BERのID,LI,EOCは必ずしも全ては必要でないため、パックト符号化規則(PER)は、BERの符号化データから予測可能なID,LI,EOCを省略する規則として定義される。例えば、明示的なタグが付加された型(型Aの要素bが参照する型B)のIDや、前の要素も自身も存在が必須のSEQUENCEの要素(型Aの要素b)のIDは省略できる。また、部分型により長さが指定されたストリング型(型B)はLIが省略できる。図1において、破線で示したLI,ID,EOCはPERで符号化した時に省略されるものである。

2.2 基本方針

PER,BER対応のC言語のプログラムを生成する方式を、以下の基本方針に従って検討した。

(1) BER+PERに対応

現在、OSIソフトウェアはBERを用いて通信することを前提として実装されており、今後PERだけを提供することは考えにくい。従って、PER単独のプログラムを生成する必要は無く、(a)BERだけ、(b)BER,PER双方を提供するプログラムを生成すればよい。(a)のコンパイラは既に存在するので[1, 5]、以下では(b)の形式だけ考慮する。

(2) 冗長性の少ないプログラム

(b)のプログラムを符号化規則毎に生成すると、双方の符号化規則に重複した処理があるためプログラムが冗長になる。2つの符号化規則の共通部分を抽出して、共通部分から規則毎にプログラムを生成するのを避け冗長さを減らす。

(3) 高速なプログラム

省略すべきIDの検出等は、符号化/復号ルーチンの実行時でなくASN.1の入力仕様から静的に解析できる。これらの情報は全てコンパイラがプログラムに変換し、高速な符号化/復号を可能とする。

3. 生成するプログラム

BER,PER対応のプログラムは、(i)ID,LIの作成/検査等の全ての型に共通なライブラリルーチン群、(ii)IDアレイ(IDに関する情報)の集合、(iii)単純型の符号化/復号ルーチン群、(iv)構造型の符号化/復号ルーチン群により実現できる。この内、(ii)と(iv)をコンパイラが生成する。BER,PER毎に別々に各ルーチンを準備することも考えられるが、2つの規則に重複する部分があるため、それぞれ1つのルーチンで実現する。具体的には、各ルーチンは呼び出し側の指定に従って、実行時にBERとPERを切替えて処理を行なう。

IDに比較してLI,EOCの省略は簡単なので、以下では主にIDの省略を中心に、図2を例にして、コンパイラが生成するプログラムについて説明する。

3.1 IDの処理

(1) IDアレイ(ID情報) IDアレイは定数で、オクテット列に符号化されたIDを保持する。このオクテット列は、明示的に付加されたタグ(2.1節参照)と型があらかじめ持つタグ(例えばIA5STRING型は[UNIVERSAL 22])から決定できる。符号化/復号ルーチンの引数になり、IDの作成/解析に使用する。ASN.1で新たに定義された型および構造型の要素毎にコンパイラが生成し、

"Design of ASN.1 Compiler for the Packed Encoding Rules"
Toru Hasegawa and Shingo Nomura
KDD R & D Laboratories

図2の例では、新しく定義された型A, B及び構造型Aの要素b, cの4つに対して生成する。IDアレイはBERとPER双方の規則に対して必要であり、双方の規則のオクテット列を1つのIDアレイに持たせる。例えば図2で、型Aの要素bのIDアレイのBER部は、その要素が持つタグ[0], [UNIVERSAL 22]から変換されたオクッテト列を保持している。このIDアレイにより、符号化/復号処理で処理回数の多いID処理を、IDアレイのオクテットと比較するだけの軽い処理にできる。

(2) PERのIDアレイ生成 PERでIDが省略されるかどうかは、入力ASN.1仕様の解析時に(静的に)わかるので、コンパイラが検出して、そのIDを省いたIDアレイを生成する。これにより、符号化/復号ルーチンは、PERの実行時に省略するIDを計算する必要がなく、高速なID処理が可能となる。

3.2 符号化ルーチン

本節では、主に図2のSEQUENCE型から生成される符号化ルーチンについて説明する。復号ルーチンも同様に生成できる。

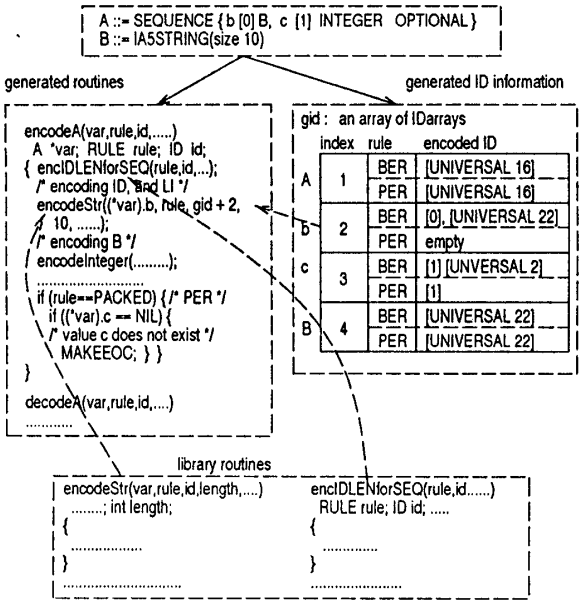

図 2: 生成プログラム

符号化ルーチンは符号化規則の種別を示す引数ruleを持ち、他のルーチンの呼び出し時にそのまま渡して、子、孫の要素のルーチンに伝搬させる。SEQUENCE型のID,LIを符号化するライブラリルーチンencIDLENforSEQを呼びだした後、構成要素のルーチンを順に呼び出す。SEQUENCEのLIは不要なので、ルーチンencIDLENforSEQは、ruleがPERを示す時LIを省略する。型Aの要素bにIDが不要なことは、コンパイラが生成するIDアレイの配列(gid)の2番めのPER部が示している。このIDアレイが渡されて、ストリングの符号化ルーチンencodeStrは要素bのIDを省略する。

SEQUENCE型において最後の要素が必須でなく、その前の要素が必須の場合、EOCを省略するかどうかは、最後の要素の値が実際に存在するかどうかに依存し、コンパイル時に決定できない。図2のルーチンencodeAの最後が示すように、規則がPERで、最後の要素cが存在しない場合だけ、EOCを省略しない(MAKEEOC)コードを生成する。

また、型Bのように部分型で長さがあらかじめ定められたストリング型のLIは省略する。そこで、ストリングの符号化ルーチンencodeStrは、その値を指定する引数lengthが非負整数を持つ場合LIを省略する。

4. 考察

基本およびバックト符号化規則を同時に提供する符号化/復号プログラムを生成する方式を検討する中で以下の考察を得た。

(1) BER用コンパイラの拡張による実現

提案したプログラム構成は、先に作成したASN.1コンパイラ[1]が生成するBERのプログラムと類似しており、BER用コンパイラを拡張して本方式のコンパイラを実現できる見通しを得た。これは、2つの規則で異なる処理をプログラムで実現するのではなく、プログラムとは独立な定数(IDアレイ)として実現したためである。また、PERがBERに対する制約で、BERで符号化したデータから不要なものを省略するという形式で定義されることも一因である。

(2) 高速なPER符号化

PERを用いても、符号化オクテット列の中で大きな割合を占めるCOは省略できないため、BERのオクテット列に比較して大きく圧縮することは期待できない。しかし、SET,CHOICE型以外では省略できるID,LI,EOCも多く、符号化/復号処理で大きな割合を占めるID,LIの作成/検査の回数を減少させられる。従って、BERに比較してかなり高速な符号化/復号が期待できる。提案したプログラムは、2つの規則の切替をコンパイル時でなく実行時に行なうが、この切替えはC言語のif文に対応するだけであり処理速度への影響は少ないと考えている。

5. まとめ

本稿では、BER,PERを同時に提供するコンパイラを提案した。ただし、PERは標準化の途中で不備な点もあり、修正が望まれる。SEQUENCE OFの要素のストリングのIDの省略がその一例である。ストリングが不定長/固定長どちらの形式かに依存してIDが省略されるので、復号時に省略されているかどうかわからない。今後は、PERの標準化の進捗にあわせて、コンパイラの設計、実装を進める予定である。さらに、実装を通して、生成したPERの符号化/復号プログラムがどの程度BERより高速であるか、双方の規則を同時に実現した場合の処理速度への影響等を明かにしたい。

最後に日頃御指導頂くKDD研究所 小野所長、浦野次長、ならびにバックト符号化規則特性に関する討論に参加して下さったOSI通信グループ堀内氏に感謝します。

文献

[1] 長谷川, 他 "ASN.1支援ツールの設計 - コンパイラおよびエディタ," 情処研究会 DPS 39-4, Sept. 1988

[2] 堀内, 他 "OSI応用層プロトコル用ASN.1圧縮符号化規則の圧縮特性に関する一考察," 情処研究会, DPS 50-3, May 1991

[3] ISO/IEC IS 8824 / 8825

[4] ISO/IEC CD 8825-2

[5] G. Neufeld, et al. "An ASN.1 to C Compiler," IEEE Trans. SE Vol.16, No.10, Oct. 1990