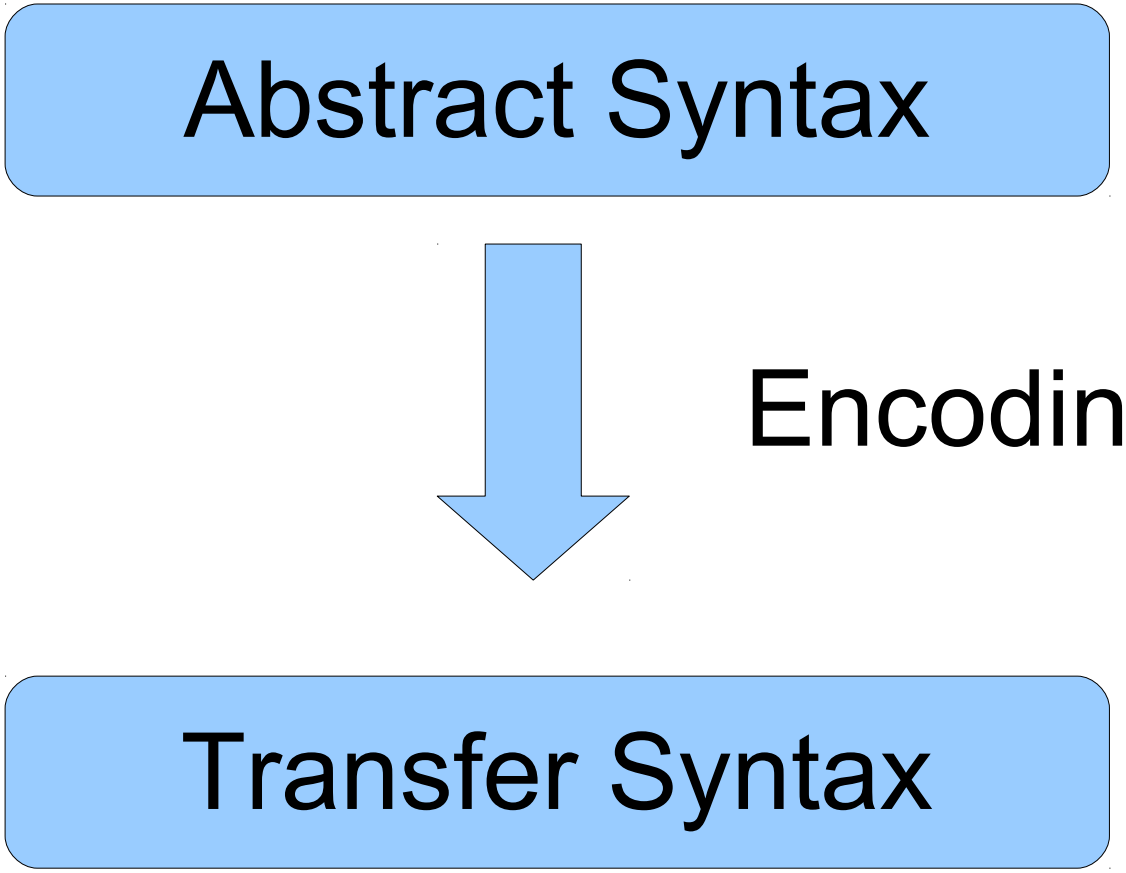
Abstract Syntax Notation One

- Język specyfikacji typów danych – np. struktury PDU
- Kilka zastosowań:
  - X.400 - katalogi LDAP
  - X.500 – certyfikaty X.509
  - RSA PKCS#12 – przechowywanie/transmisja kluczy i certyfikatów
  - H.323 – wymiana informacji o możliwościach terminala
  - SNMP – MIB oraz PDU

Model reprezentacji danych



Struktura definicji ASN.1

Moduł: kolekcja definicji typów i wartości

```
NazwaModulu { 1 0 6 } DEFINITIONS ::=
BEGIN
    TypX ::= SEQUENCE {
        A INTEGER
    }
    WartoscY  INTEGER ::= 4
END
```

Typy proste

- BOOLEAN, INTEGER, REAL
- ENUMERATED
- BIT STRING , OCTET STRING
  - '11010001'B
  - '82DA6'H
- CHARACTER STRING (NumericString, PrintableString, UTF8String)
- UTCTime, GeneralizedTime
- NULL
- OBJECT IDENTIFIER

Zawężenia typów

<subtype name> ::= <type> (<constraint>)

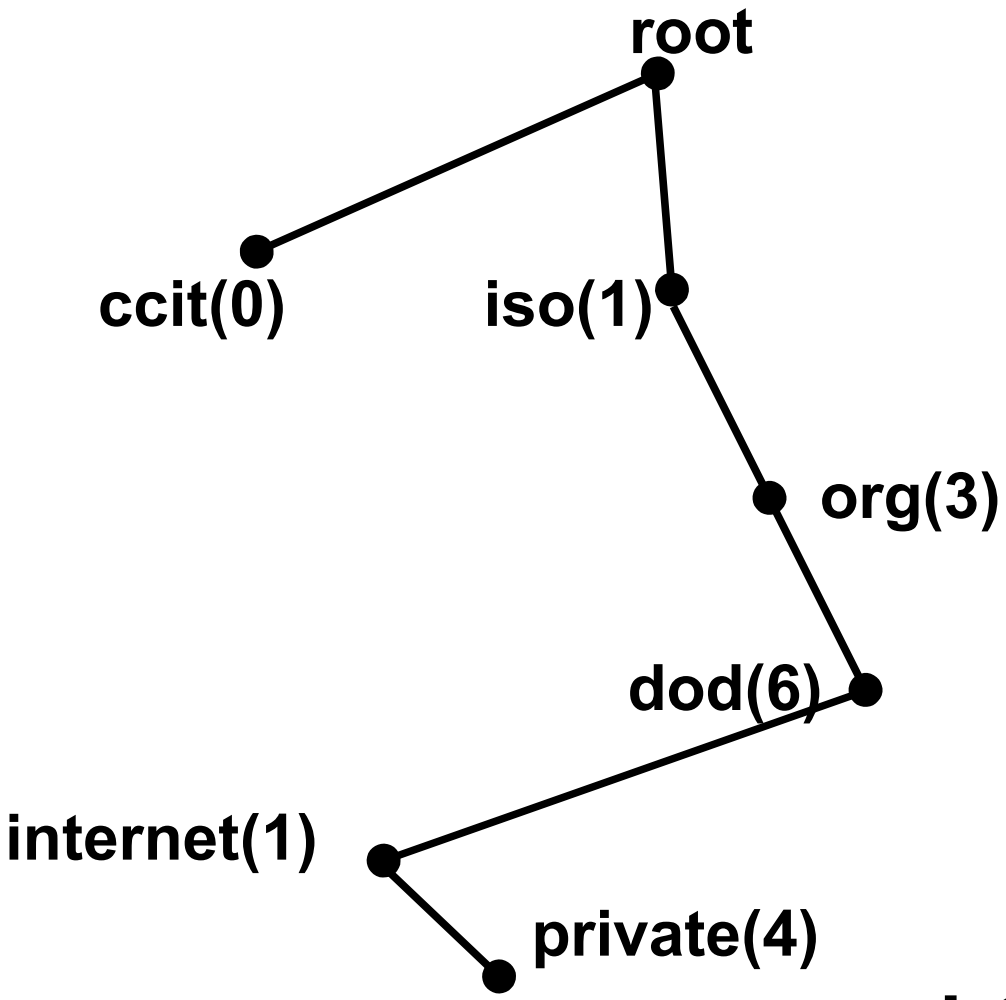
Counter ::= INTEGER (0..4294967295)
IpAddress ::= OCTET STRING (SIZE(4))
Spring ::= Months (march | april | may)
Summer ::= Months (june | july | august)
SmallPrime ::= INTEGER (2 | 3 | 5 | 7 | 11)
ExportKey ::= BIT STRING (SIZE(40))

Przypisania wartości

<value name> <type> ::= <value>

ipInReceives Counter ::= 2450
ipRouteMask IpAddress ::= 'FFFFFF00'H
currentMonth Months ::= february
currentTime UTCTime ::= "000204075015+0100"
givenName VisibleString ::= "Andreas"
married BOOLEAN ::= TRUE
faxMessage BIT STRING ::= '01100001101'B
encryptionKey ExportKey ::= 'A1B2C3D4E5'H

Object Identifier



SNMP

IBM: { enterprise 2 }
Cisco: { enterprise 9 }
Hewlett-Packard: { enterprise 11 }
Sun Microsystems: { enterprise 42 }
Microsoft: { enterprise 311 }
Intel: { enterprise 343 }

http://www.oid-info.com/

Typy złożone

- Sekwencja (SEQUENCE) – kolekcja pól różnego typu, których kolejność jest istotna.

```
Definicja typu: UserAccount ::= SEQUENCE {
                    Username VisibleString,
                    Password Visible String,
                    AccountNr INTEGER
                }

Wartość: myAccount UserAccount ::= {
                     username      "steffen",
                     password         "jane51",
                     accountNr        4711
                }
```

Typy złożone

- Sekwencja2 (SEQUENCE OF) – kolekcja pól tego samego typu, których kolejność jest istotna.

```
MemberCountries ::= SEQUENCE OF VisibleString
AccountRegistry ::= SEQUENCE OF UserAccount

euro2012hosts MemberCountries ::= {
   ”Poland”,”Ukraine”
}
```

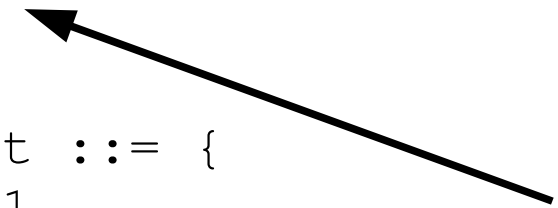
- Analogicznie SET OF (patrz następny slajd)

Typy złożone

- Zbiór (SET) – kolekcja pól różnego typu, których kolejność jest nieistotna.

```
UserAccount ::= SET {
        username [0] VisibleString,
        password [1] VisibleString,
        accountNr [2] INTEGER
}

myAccount UserAccount ::= {
      accountNr  4711,
      username "johny",
      password "ALAMAKOTA"
}
```



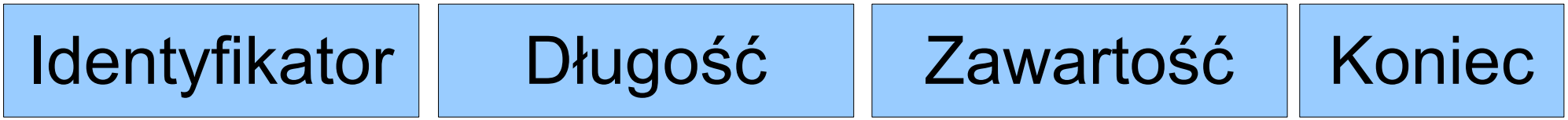
Context-specific tag

Encoding Rules

- Basic Encoding Rules (BER)
- Canonical i Distinguished (CER,DER) – BER z nałożonymi restrykcjami = deterministyczne
- Packed encoding rules (PER) – optymalizacja rozmiaru
- XML Encoding Rules (XER)
- Generic String Encoding Rules (GSER)

Basic Encoding Rules

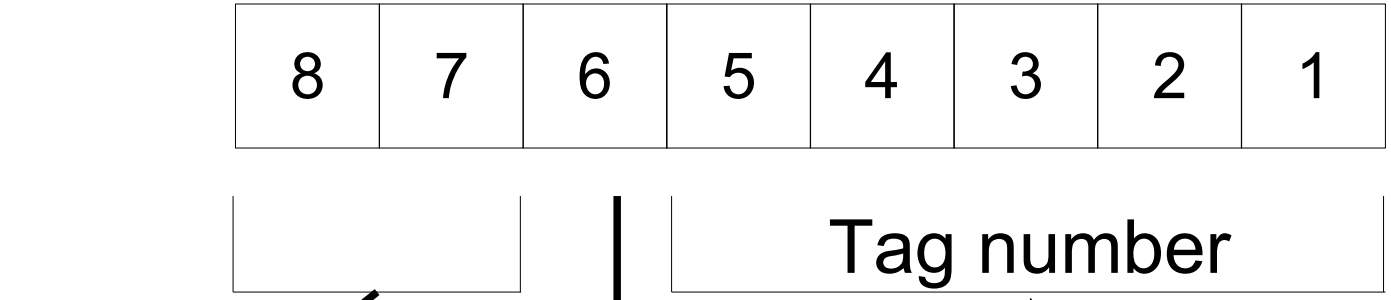
- Trzy metody kodowania:
  - primitive, definite-length encoding
  - constructed, definite- length encoding
  - constructed, indefinite-length encoding



- Znaczniki typu:
  - Universal: BOOLEAN(1), INTEGER(2), BIT STRING(3),OCTET STRING(4),OID(6), REAL(9), SEQUENCE(16)
  - Application

```
Counter ::= [APPLICATION 1]  INTEGER
```

Identyfikator pola



0 – Primitive
1 – Constructed

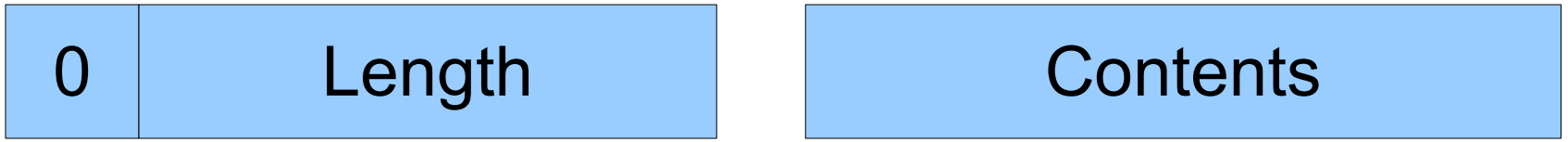
Klasa
00 – Universal
01 – Application
10 – Context-specific
11 - Private

Jeśli numer > 31

Ustawiamy wartość tego pola na 11111 a następnie kodujemy w kolejnych oktetach zaczynając każdy z nich 1 a w ostatnim zapisując 0

Pole długości

- Short definite form (długość < 128 oktetów)



- Long definite form (128 ≤ długość < 21008 oktetów)

- Indefinite form

Dwa oktety 0