

ABNF (cont)

```
alternation = concatenation
             >(*c-wsp "/" *c-wsp concatenation)
concatenation = repetition *(1*c-wsp repetition)
repetition = [repeat] element
repeat = 1*DIGIT / (*DIGIT "*" *DIGIT)
element = rulename / group / option /
          char-val / num-val / prose-val
```

- ◊ Alternatives
 - A / B
 - Either A or B (not both)
- ◊ Sequence
 - A B C
 - A followed by B followed by C
- ◊ Repetition
 - m*nA or pA
 - a sequence of A's
 - at least m no more than n
 - defaults for m and n: 0 and infinity
 - or exactly p A's

ABNF (cont)

```
element = rulename / group / option /
          char-val / num-val / prose-val
group = "(" *c-wsp alternation *c-wsp ")"
option = "[" *c-wsp alternation *c-wsp "]"
char-val = DQUOTE *(%x20-21 / %x23-7E) DQUOTE
           ; quoted string of SP and VCHAR
           ; without DQUOTE
num-val = "%" (bin-val / dec-val / hex-val)
bin-val = "b" 1*BIT
           [ 1*("." 1*BIT) / ("-" 1*BIT) ]
           ; series of concatenated bit values
           ; or single ONEOF range
dec-val = "d" 1*DIGIT
           [ 1*("." 1*DIGIT) / ("-" 1*DIGIT) ]
hex-val = "x" 1*HEXDIG
           [ 1*("." 1*HEXDIG) / ("-" 1*HEXDIG) ]
prose-val = "<" *(%x20-3D / %x3F-7E) ">"
            ; bracketed string of SP and VCHAR
            ; without angles
            ; prose description, to be used as
            ; last resort
```

- ◊ That is it.
 - other than a few more core rules

ABNF Examples - from rfc2822

```
atom = [CFWS] 1*atext [CFWS]
atext = ALPHA / DIGIT / "!" / "#" / "$" /
        "%" / "&" / "'" / "*" / "+" / "-" /
        "/" / "=" / "?" / "^" / "_" / "`" /
        "{" / "|" / "}" / "~"
        ; Any character except
        ; controls, SP,
        ; and specials.
        ; Used for atoms
CFWS = *([FWS] comment)
comment = "(" *([FWS] ccontent) [FWS] ")"
ccontent = ctext / quoted-pair / comment
```

- ◊ Defines an atom
 - basic token of e-mail parsing
- ◊ Uses core rules from ABNF
 - (not complete above)

More E-mail

```
word      =      atom / quoted-string
phrase    =      1*word / obs-phrase

address   =      mailbox / group
mailbox   =      name-addr / addr-spec
name-addr =      [display-name] angle-addr
angle-addr =      [CFWS] "<" addr-spec ">" [CFWS]
group     =      display-name ":"
           [mailbox-list / CFWS] ";" [CFWS]

display-name = phrase
mailbox-list = mailbox *("," mailbox)
address-list = address *("," address)
```

- ◊ obs-xxx
 - obsolete syntax elements
 - retained for backwards compatibility
 - not to be generated any more
 - mostly removed from example here

E-mail Addresses

```
addr-spec = local-part "@" domain
local-part = dot-atom / quoted-string
domain     = dot-atom / domain-literal
domain-literal = [CFWS] "[" *([FWS] dcontent)
                [FWS] "]" [CFWS]
dcontent   = dtext / quoted-pair
dtext      = NO-WS-CTL /
            %d33-90 /
            %d94-126

dot-atom   = [CFWS] dot-atom-text [CFWS]
dot-atom-text = 1*atext *( "." 1*atext)
```

- ◊ kre@munnari.OZ.AU
- ◊ "just about anything"@some.domain
- ◊ user.name@[not-quite-anything]

E-MAIL

- ◊ RFC2822
 - defines the format of a message
- ◊ RFC2821
 - defines the protocol used to transmit a message

```
message = (fields / obs-fields)
         [CRLF body]

body    = *( *998text CRLF ) *998text

text    = %d1-9 /      ; Characters excluding CR and LF
         %d11 /
         %d12 /
         %d14-127 /
         obs-text
```

E-MAIL

- ◊ fields are the message header
- ◊ body is the message
 - note line length limit

```
body =>(*998text CRLF) *998text
```
 - note character set restriction

```
text = %d1-9 / %d11 / %d12 / %d14-127
```
- ◊ This all inherited from RFC822
 - Methods to enhance e-mail needed
 - So can send arbitrary data
 - So can send non-ascii characters

E-Mail Headers

```
fields = *(trace
           *(resent-date /
             resent-from /
             resent-sender /
             resent-to /
             resent-cc /
             resent-bcc /
             resent-msg-id))
          *(orig-date /
            from /
            sender /
            reply-to /
            to /
            cc /
            bcc /
            message-id /
            in-reply-to /
            references /
            subject /
            comments /
            keywords /
            optional-field)
```

Header requirements

- ◊ Grammar doesn't tell everything
 - Some fields are required
 - In grammar, everything is optional
 - Too hard to specify
 - because order is (mostly) irrelevant
 - Also want to specify occurrences
- ◊ See RFC2822 for field syntax
 - And field meanings (semantics)
- ◊ This one is interesting...

```
optional-field = field-name ":" unstructured CRLF
field-name     = 1*ftext
ftext         = %d33-57 / ; Any character except
               %d59-126 ; controls, SP, and ":".
unstructured  = *([FWS] utext) [FWS]
utext        = NO-WS-CTL / ; Non white space controls
               %d33-126 ; The rest of US-ASCII
FWS          = [*WSP CRLF] 1*WSP ; Folding white space
```

New Header Fields

- ◊ The optional-field allows new fields to be created
 - Almost anything is OK
- ◊ Little co-ordination of field names
 - RFC822
 - names starting with X for random invention
 - never be defined in a standard
 - other names for standards
 - Doesn't work.
 - Why?

Example Headers

```
Return-Path: <s4612099@maliwan.psu.ac.th>
Received: from maliwan.psu.ac.th (maliwan.psu.ac.th [202.12.73.4])
  by munnari.OZ.AU (8.12.8/8.11.6) with ESMTTP id h6PEbr20026452
  for <kre@munnari.OZ.AU>; Sat, 26 Jul 2003 00:37:57 +1000 (EST)
Received: from Santichai (santichai.std.coe.psu.ac.th [172.30.20.134])
  by maliwan.psu.ac.th (8.11.6/8.11.6) with SMTP id h6PEbo104443
  for <kre@munnari.OZ.AU>; Fri, 25 Jul 2003 21:37:50 +0700 (ICT)
Message-ID: <000e01c352ba$54e9a9a0$86141eac@Santichai>
From: "Santichai Chuaywong" <s4612099@maliwan.psu.ac.th>
To: <kre@munnari.OZ.AU>
Subject: Some mistake about lecture slide
Date: Fri, 25 Jul 2003 21:37:52 +0700
MIME-Version: 1.0
Content-Type: multipart/alternative;
  boundary="====_NextPart_000_000B_01C352F5.009425C0"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.50.4807.1700
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4910.0300
```

- ◊ X-Priority => Priority
 - Always need both