

```

rulelist      = 1*( rule / (*c-wsp c-nl) )

rule          = rulename defined-as elements c-nl
                  ; continues if next line starts
                  ; with white space

rulename      = ALPHA *(ALPHA / DIGIT / "-")

defined-as    = *c-wsp ("=" / "=/") *c-wsp
                  ; basic rules definition and
                  ; incremental alternatives

elements     = alternation *c-wsp

c-wsp        = WSP / (c-nl WSP)

c-nl         = comment / CRLF
                  ; comment or newline

comment      = ";" *(WSP / VCHAR) CRLF

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

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

```

