0. Introduction

This follows on to Murata-san's e-mails in the threads "My proposals: content type and media ypest" (last from 21 Feb 2015) and "Which RFC(s) for media type should we refer to?" (last from 11 Dec 2014) analyzing the regular expression in the schema for ST_ContentType in Part 2 Annex D. It simplifies the XML Schema regular expression and compares it to RFC 2616's definition of media-type. It then compares the definition of media-type in RFC 2616 and RFC 7231, which obsoletes RFC 2616.

Tracked changes show the progression of analyzing the regex. In at least Microsoft Word, you can click on a comment to see it highlight only the segment of BNF it relates to.

John Haug, 27 Feb 2015

1. Part 2 schema regex simplification

Original Part 2 schema regex pattern (ST_ContentType)

```
(((([\p{IsBasicLatin}-[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]])+))/((([\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]])+))((\s+)*;(\s+)*(((([\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]])+))=((([\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]])+)|(&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r]]|(\s+))|(\\[\p{IsBasicLatin}]))*&quot;))))*)
```

Replace same chunks with X (not using correct XSD entity reference notation for simplicity)

```
((((~~[\p{IsBasicLatin}-[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]]~~X)+))/((([\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]]X)+))((\s+)*;(\s+)*(((([\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]]X)+))=((([\p{IsBasicLatin}-
[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]]X)+)|(&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r]]|(\s+))|(\\[\p{IsBasicLatin}]))*&quot;))))*)
```

Remove obvious unnecessary parentheses

```
~~((((~~X~~)~~+~~))~~/~~(((~~X~~)~~+~~))~~((\s+)*;(\s+)*((~~((~~X~~)~~+~~))~~=(~~((~~X~~)~~+~~))~~|(&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;\n\r]]|~~(~~\s+~~)~~)|(\\[\p{IsBasicLatin}]))*&quot;))))*~~)~~
```

Remove other unnecessary parentheses, modifiers (+) and character specifications (\n\r covered by \p{Cc})

```
X+/X+(~~(~~\s~~+)~~*;~~(~~\s~~+)~~*(~~(~~X+~~)~~=(X+|(&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-
[\p{Cc}&#127;&quot;~~\n\r~~]]|\s+)|(\\[\p{IsBasicLatin}]))*&quot;))))*
```

Remove unnecessary parentheses, replace chunk with Y (ignore whitespace added for ease of grouping/reading)

```
X+/X+(\s*;\s*(X+=(
```

```
      X+ | (&quot;(([\p{IsLatin-1Supplement}\p{IsBasicLatin}-[\p{Cc}&#127;&quot;]]|\s+)Y | (\\[\p{IsBasicLatin}])*)&quot;)
)))*
```

Simplify (remove added whitespace)

```
X+/X+(\s*;\s*(X+=(X+|(&quot;(Y|(\\[\p{IsBasicLatin}])*)&quot;))))*
```

2. Definitions #1 – RFC 2616 and regex

RFC 2616

NOTE: This uses old RFC 822-style augmented Backus-Naur Form (not the same as RFC 5234 ABNF)

```
media-type     = type "/" subtype *( ";" parameter )
type           = token
subtype        = token
parameter      = attribute "=" value
attribute      = token
value          = token | quoted-string
quoted-string  = ( <"> *(qdtext | quoted-pair ) <"> )
qdtext         = <any TEXT except <">>
quoted-pair    = "\" CHAR
TEXT           = <any OCTET except CTLs, but including LWS>
LWS            = [CRLF] 1*( SP | HT )   ; linear white space
CRLF           = CR LF
token          = 1*<any CHAR except CTLs or separators>
CHAR           = <any US-ASCII character (octets 0 - 127)>
CTL            = <any US-ASCII control character (octets 0 - 31) and DEL (127)>
separators     = "(" | ")" | "<" | ">" | "@" | "," | ";" | ":" | "\" | <"> | "/" | "[" | "]" | "?" | "=" | "{" | "}" | SP | HT
OCTET          = <any 8-bit sequence of data>
CR             = <US-ASCII CR, carriage return (13)>
LF             = <US-ASCII LF, linefeed (10)>
SP             = <US-ASCII SP, space (32)>
HT             = <US-ASCII HT, horizontal-tab (9)>
<">            = <US-ASCII double-quote mark (34)>
```

Regex

From http://www.regular-expressions.info/unicode.html:

`\p{Cc}` or `\p{Control}`: an ASCII 0x00–0x1F or Latin-1 0x80–0x9F control character

Commented [JH1]: \p{IsBasicLatin}

Commented [JH2]: \p{Cc}&#127;

Commented [JH3]: \p{IsBasicLatin}\p{IsLatin-1Supplement}

From https://msdn.microsoft.com/en-us/library/20bw873z(v=vs.110).aspx#SupportedNamedBlocks:

Code point range	Block name	Note
0000 - 007F	IsBasicLatin	ASCII 0-127
0080 - 00FF	IsLatin-1Supplement	Extended ASCII 128-255

3. Interpretation of X and Y in Part 2 regex

```
X: [\p{IsBasicLatin}-[\p{Cc}&#127;\(\)&lt;&gt;@,;:\\&quot;/\[\]\?=\{\}\s\t]]
```

- English approximation: Any single ASCII character except controlchars DEL ( ) < > @ , ; : \ " / [ ] ? = { } SPACE TAB

X+ is the same as RFC 2616's token

```
Y: ([\p{IsLatin-1Supplement}\p{IsBasicLatin}-[\p{Cc}&#127;&quot;]]|\s+)
```

- English approximation: Any single extended ASCII character including linear whitespace except controlchars DEL "

Y is the same as RFC 2616's qdtext

4. Comparison of simplified Part 2 schema regex and RFC 2616 media-type

```
media-type = type "/" subtype *( ";" parameter )

           = token "/" token *( ";" attribute "=" value)

           = token "/" token *( ";" token "=" (token | quoted-string))

           = token "/" token *( ";" token "=" (token | (<"> *(qdtext | quoted-pair) <">)))

           = token "/" token *( ";" token "=" (token | (<"> *(qdtext | "\" CHAR) <">)))
```

```
X+/X+(\s*;\s*(X+=(X+|(&quot;(Y|\\[\p{IsBasicLatin}]*)&quot;))))*
```

5. Definitions #2 – RFC 7231

RFC 7231

NOTE: This uses RFC 5234 Augmented Backus-Naur Form

```
media-type    = type "/" subtype *( OWS ";" OWS parameter )
type          = token
subtype       = token
```

Commented [JH4]: CHAR

Commented [JH5]: CTL

Commented [JH6]: separators

Commented [JH7]: type

Commented [JH8]: subtype

Commented [JH9]: attribute

Commented [JH10]: value

Commented [JH11]: type

Commented [JH12]: subtype

Commented [JH13]: OPC allows whitespace around the ; which is different from RFC 2616

\s allows [\t\r\n\f] plus possibly vertical tab and possibly Unicode "separators"

RFC 7231 allows only [\t]

Commented [JH14]: attribute

Commented [JH15]: token

Commented [JH16]: qdtext

Commented [JH17]: quoted-pair

Commented [JH18]: quoted-string

Commented [JH19]: value

```
parameter     = token "=" ( token / quoted-string )
quoted-string = DQUOTE *( qdtext / quoted-pair ) DQUOTE
qdtext        = HTAB / SP / %x21 / %x23-5B / %x5D-7E / obs-text
quoted-pair   = "\" ( HTAB / SP / VCHAR / obs-text )
token         = 1*tchar
tchar         = "!" / "#" / "$" / "%" / "&" / "'" / "*" / "+" / "-" / "." / "^" / "_" / "`" / "|" / "~" / DIGIT / ALPHA
                ; any VCHAR, except delimiters
VCHAR         = %x21-7E             ; visible (printing) characters, RFC 5234 Appendix B.1
DIGIT         = %x30-39             ; RFC 5234 Appendix B.1
ALPHA         = %x41-5A / %x61-7A   ; A-Z / a-z
obs-text      = %x80-FF
OWS           = *( SP / HTAB )      ; optional whitespace, RFC 7230 Section 3.2.3
SP            = %x20                ; RFC 5234 Appendix B.1
HTAB          = %x09                ; horizontal tab, RFC 5234 Appendix B.1
```

6. Differences between RFC 2616 and RFC 7231

media-type

RFC 2616 disallows whitespace around the semi-colon preceding a parameter. RFC 7231 allows any number of SP and/or HTAB.

token

No differences

qdtext

RFC 2616 includes LF (octet 10 / %x0A), CR (octet 13 / %x0D), \ (octet 92 / %x5C). RFC 7231 disallows these characters.

quoted-pair

RFC 2616 allows any character in the standard ASCII range (octets 0-127). RFC 7231 disallows the range octets 0-31 except for octet 9 (HTAB).