**Grammar (Proposed Revised)**

Addresses issues with use of word “representation” too loosely.

Aligns more with the description of how things have to work, as Nils must be checked early. (per action 140 stuff)

Breaks out overused Initiator and Terminator regions into Nil, Empty, Sequence, Choice, and Element variants. To me this makes things much clearer that the initiator of a sequence is NOT the one the element specifies, but must be specified on the sequence itself.

**Updated by Steve:**

Use ‘Rep’ in a manner consistent with action 140 definitions

Use ‘Content’ consistently

Separate simple and complex so that empty v normal ordering can be correctly shown

Element f

raming and content cells merged

Explicitly include empty value

Not sure why we have separate Initiator and Terminator regions. This is not consistent with other properties shared between elements, sequences and choices.

**Updated by Mike: Changes based on 2012-07-25 WG Call**

Eliminated SequenceInitiator/Terminator, Choice, Element variants on these also.

Refactored Left/Right Framing. Made Empty left/right framing productions for parallel structure

.

Removed ***EmptyValue*** terminal from grammar. Was confusing additional use of Empty where we did NOT mean EmptyElement. This meant “no data”. Right way to say that in a grammar is to have nothing at all. (However, we have this ***Absent*** terminal....?? We need one way in the grammar to say that there are no representation bits.)

Changed Empty to EmptyElement, Nil to NilElement

**Updated by Steve:**

Removed EmptyElementRep for symmetry.

Added ‘Element’ into AbsentRep to match Empty and Nil.

Corrected EmptyElementRepXxxxFraming to EmptyElementXxxxFraming

Made NilElementLiteralCharacter plural so NilElementLiteralCharacter**s**

Personally I think we need ***EmptyValue***, because we have ***Absent***

I think NilElementLiteralValue, NilElementLogicalValue and NilElementLiteralCharacters should drop the ‘Element’, we are talking about values now so the qualification is not needed.

MikeB (7/31)

Shortened NilElementLiteralValue et al. Per steve's last comment above.

A point of clarification: we need Absent because it is part of an alternative composition. One only can remove zero-length terminals if they are in sequential compositions.

Mikeb (2nd round of changes on 7/31)

Modified to disallow top level elements to be arrays, and top level elements to be Absent.

Only group/array contents can be absent. So I generalized and expanded the notion of Enclosed… so that we distinguish between a DocumentElement – which cannot be absent, and an EnclosedElement, which can be Absent.

TBD: Do we have to distinguish between SimpleAbsentElementRep and ComplexAbsentElementRep? Could these just be AbsentElementRep or do we really want both productions so that we can discuss them explicitly?

Mikeb (2012-08-07) Collapsed SimpleAbsentElementRep and ComplexAbsentElementRep to just AbsentElementRep

Mikeb(2012-11-19) SequenceContent missing final “]” in production.

|  |  |
| --- | --- |
| **Notes (not for final spec perhaps)** | **Productions** |
| Top Level | Document = ***UnicodeByteOrderMark*** DocumentElementDocumentElement = SimpleElement | ComplexElementEnclosedElement = SimpleEnclosedElement | ComplexEnclosedElementSimpleElement = SimpleLiteralNilElementRep | SimpleEmptyElementRep |  SimpleNormalRep SimpleEnclosedElement = SimpleElement | AbsentElementRep ComplexElement = ComplexLiteralNilElementRep | ComplexNormalRep |  ComplexEmptyElementRep ComplexEnclosedElement = ComplexElement | AbsentElementRep  |
| Absent Element | AbsentElementRep = ***Absent*** |
| Empty Element | SimpleEmptyElementRep = EmptyElementLeftFraming EmptyElementRightFramingComplexEmptyElementRep = EmptyElementLeftFraming EmptyElementRightFramingEmptyElementLeftFraming = LeadingAlignment ***EmptyElementInitiator*** PrefixLengthEmptyElementRightFraming = ***EmptyElementTerminator*** TrailingAlignment |
| Literal Nil Element | SimpleLiteralNilElementRep = NilElementLeftFraming [***NilLiteralCharacters*** |  NilElementLiteralContent] NilElementRightFramingComplexLiteralNilElementRep = NilElementLeftFraming NilElementRightFramingNilElementLeftFraming = LeadingAlignment ***NilElementInitiator*** PrefixLengthNilElementRightFraming = ***NilElementTerminator*** TrailingAlignmentNilElementLiteralContent = ***LeftPadding*** ***NilLiteralValue*** RightPadOrFill |
| Normal Element | SimpleNormalRep = LeftFraming PrefixLength SimpleContent RightFramingComplexNormalRep = LeftFraming PrefixLength ComplexContent ***ElementUnused***RightFramingLeftFraming = LeadingAlignment ***Initiator*** RightFraming = ***Terminator*** TrailingAlignmentPrefixLength = SimpleContent | PrefixPrefixLength SimpleContentPrefixPrefixLength = SimpleContent SimpleContent =  ***LeftPadding*** [ ***NilLogicalValue | SimpleValue*** ]RightPadOrFillComplexContent = Sequence | Choice |
| Complex Content Regions | Sequence = LeftFraming SequenceContent RightFraming SequenceContent = [ ***PrefixSeparator***EnclosedContent [ ***Separator*** EnclosedContent ]\*  ***PostfixSeparator*** ]Choice = LeftFramingChoiceContent RightFramingChoiceContent = [ EnclosedContent ] ***ChoiceUnused***EnclosedContent = [ EnclosedElement | Array | Sequence | Choice ]Array = [ EnclosedElement [ ***Separator*** EnclosedElement ]\* [ ***Separator*** StopValue] ]StopValue = SimpleElement |
| Details of Other Framing | LeadingAlignment = ***LeadingSkip*** ***AlignmentFill***TrailingAlignment = ***TrailingSkip***RightPadOrFill = ***RightPadding*** | ***RightFill*** |

Notes:

* For a simple element, if nilKind is ‘literalCharacter’ the nil comparison occurs before trimming when parsing, otherwise it occurs after trimming. See errata 2.8. *This is modelled explicitly in the grammar now. So this note isn’t needed.*