

Extrait des règles BNF d'un document XML

[1]	document	::= <u>prolog</u> <u>element</u> <u>Misc</u> *
[2]	Char	::= #x9 #xA #xD [#x20-#xD7FF] [#xE000-#xFFFD] [#x10000-#x10FFFF]
[3]	S	::= (#x20 #x9 #xD #xA)+
[4]	NameStartChar	::= ":" [A-Z] "_" [a-z] [#xC0-#xD6] [#xD8-#xF6] [#xF8-#x2FF] [#x370-#x37D] [#x37F-#x1FFF] [#x200C-#x200D] [#x2070-#x218F] [#x2C00-#x2FEF] [#x3001-#xD7FF] [#xF900-#xFDCF] [#xFDF0-#xFFFD] [#x10000-#xEFFFF]
[4a]	NameChar	::= <u>NameStartChar</u> "-" "." [0-9] #xB7 [#x0300-#x036F] [#x203F-#x2040]
[5]	Name	::= <u>NameStartChar</u> (<u>NameChar</u>)* <i>Name ::= NOM.</i>
[10]	AttValue	::= "'" ([^&"] <u>Reference</u>)* "'" '"' ([^&'] <u>Reference</u>)* '"'
[14]	CharData	::= [<&]* - ([^<&]* '']>' [<&]*
[15]	Comment	::= '<!--' ((Char - '-') ('-' (Char - '-')))* '-->'
[16]	PI	::= '<?' <u>PI</u> Target (<u>S</u> (<u>Char</u> * - (<u>Char</u> * '']>' <u>Char</u> *) ^{Attribut})* '??' '??' → on va l'utiliser de le car du prolog seulement
[17]	PITarget	::= <u>Name</u> - (('X' 'x') ('M' 'm') ('L' 'l'))
[18]	CDSect	::= <u>CDStart</u> <u>CData</u> <u>CDEnd</u>
[19]	CDStart	::= '<![CDATA['
[20]	CData	::= (<u>Char</u> * - (<u>Char</u> * '']>' <u>Char</u> *)
[21]	CDEnd	::= ']]>'
[22]	prolog	::= <u>XMLDecl</u> ? <u>Misc</u> * (<u>doctypedecl</u> <u>Misc</u> *)?
[23]	XMLDecl	::= '<?xml' <u>VersionInfo</u> <u>EncodingDecl</u> ? <u>SDDecl</u> ? <u>S</u> ? '??>'
[24]	VersionInfo	::= <u>S</u> 'version' <u>Eq</u> ("'" <u>VersionNum</u> "'" "'" <u>VersionNum</u> "'')
[25]	Eq	::= <u>S</u> ? '=' <u>S</u> ?
[26]	VersionNum	::= '1.' [0-9]+
[27]	Misc	::= <u>Comment</u> <u>PI</u> <u>S</u> - <i>PI: Processing Instruction</i>
[28]	doctypedecl	::= '<!DOCTYPE' <u>S</u> <u>Name</u> (<u>S</u> <u>ExternalID</u>)? <u>S</u> ? '[' <u>IntSubset</u> ']' <u>S</u> ? '??>' → pas de DTD!
[39]	element	::= <u>EmptyElemTag</u> <u>STag</u> <u>content</u> <u>ETag</u> <i>Descript d'un élém XML en Bison</i>
[40]	STag	::= '<' <u>Name</u> (<u>S</u> <u>Attribute</u>)* <u>S</u> ? '>'
[41]	Attribute	::= <u>Name</u> <u>Eq</u> <u>AttValue</u>
[42]	ETag	::= '</' <u>Name</u> <u>S</u> ? '>'
[43]	content	::= <u>CharData</u> ? ((<u>element</u> <u>Reference</u> <u>CDSect</u> <u>PI</u> <u>Comment</u>) <u>CharData</u> ?)*
[44]	EmptyElemTag	::= '<' <u>Name</u> (<u>S</u> <u>Attribute</u>)* <u>S</u> ? '/>'

→ espace; mais ça nous sert pas