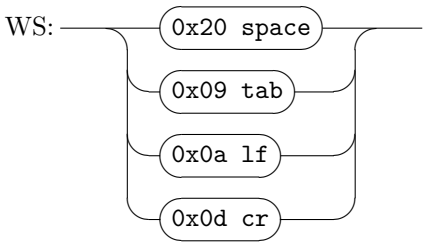
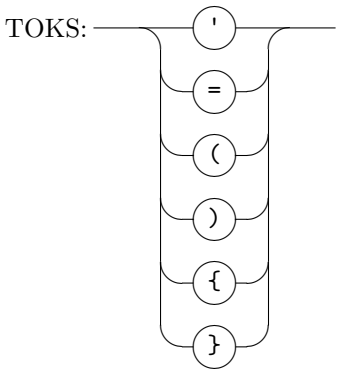
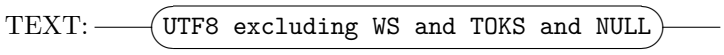
Specification for XMQ by Fredrik Öhrström 2021-01-19 oehrstroem@gmail.com 1/3

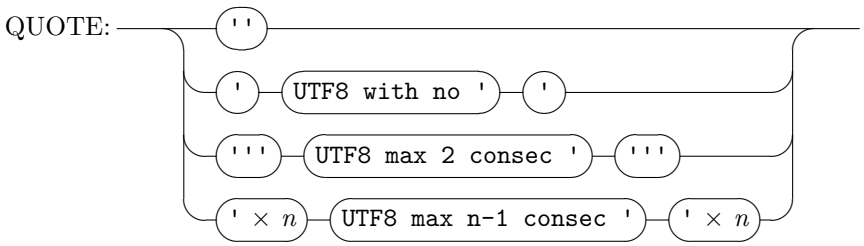
An xmq file must be valid UTF8 and any CRLF is changed into LF before lexing/parsing.

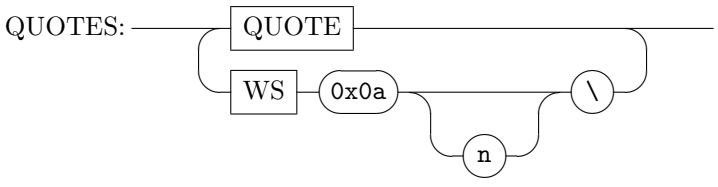
Lexer

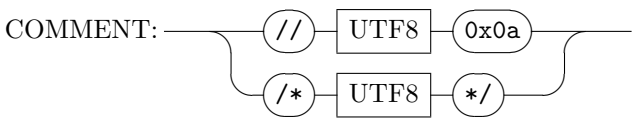


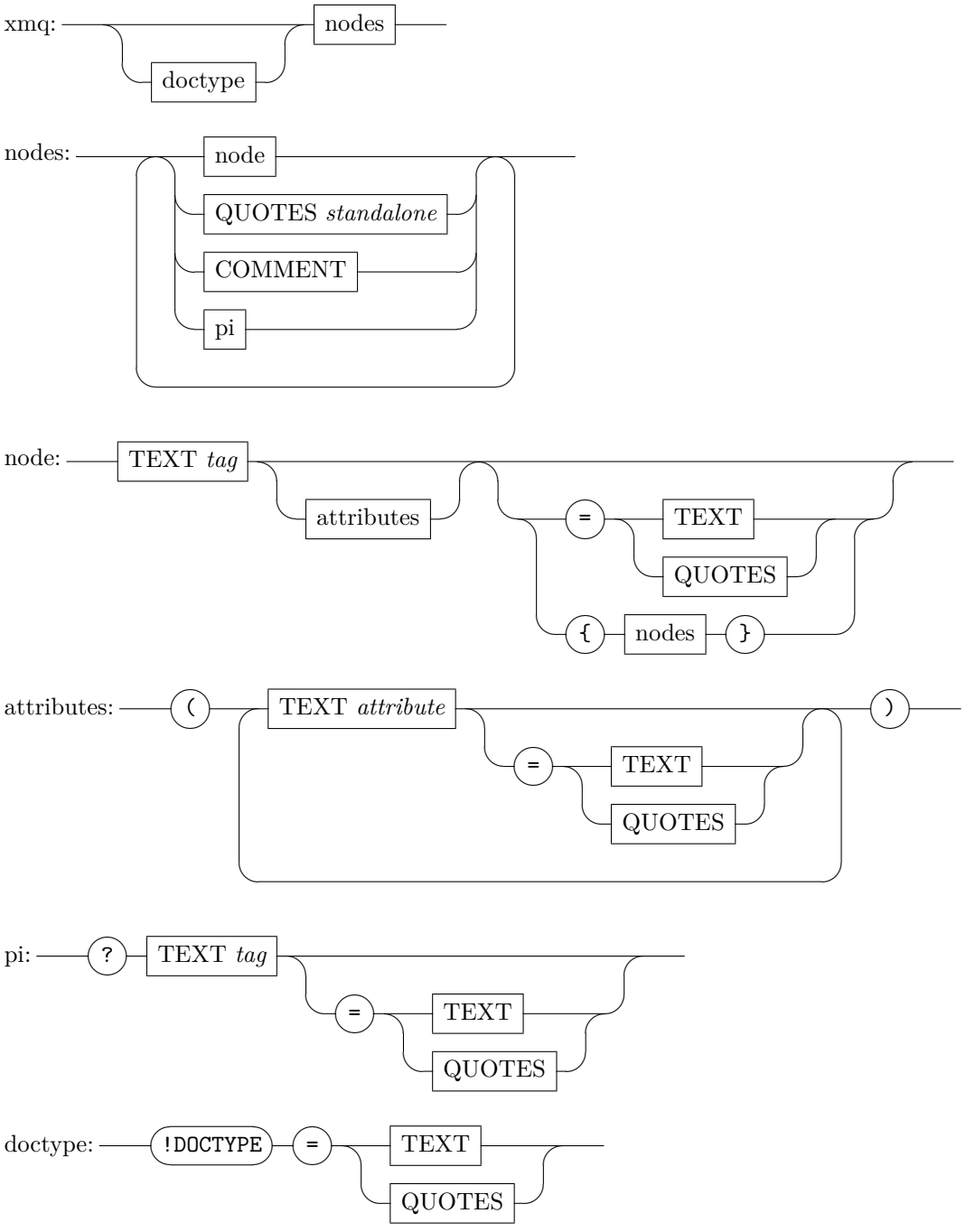











xmq:
nodes
doctype
nodes:
node
QUOTES standalone
COMMENT
pi
node:
TEXT tag
attributes
=
TEXT
QUOTES
{
nodes
}
attributes:
(
TEXT attribute
=
TEXT
QUOTES
)
pi:
?
TEXT tag
=
TEXT
QUOTES
doctype:
!DOCTYPE
=
TEXT
QUOTES

TEXT tag and attribute must conform to the XML element naming rules:

1. must start with a letter or underscore
2. cannot start with the letters xml (or XML, or Xml, etc)
3. can contain letters, digits, hyphens, underscores, and periods

QUOTE content is trimmed according to these rules:

1. Any leading/ending (space|cr|tab)* nl (space|cr|tab)* is removed.
2. Incidental whitespace will always be trimmed, when there is at least one nl left.

QUOTES can combine multiple QUOTE(s):

1. A suffix \ joins the next QUOTE seamlessly.
2. A suffix \n inserts a newline before joining.

Joining children inside the nodes rule:

1. All children are joined seamlessly.
2. Except, two consecutive standalone QUOTEs are joined with a newline.

Pretty printing introduces whitespace for div{a{span=' hi '}}

1. Default XML output will pretty print and insert newlines+indentation.
   This is usually safe, use --nopp to not pretty print.

```
<div>
    <a>
        <span> hi </span>
    </a>
</div>
```

2. Default HTML output is without pretty printing since
   introducing newlines+indentation in HTML requires DTD/CSS knowledge
   where it is possible to safely introduce whitespace.

```
<div><a><span> hi </span></a></div>
```

3. It is possible to pretty print HTML with --pp and get a best effort guess of
   where it is safe to introduce newlines+indentation.

```
<div>
    <a><span> hi </span></a></div>
```