Algorithm 1 Initial state and value declarations (Part I)

```
Require:
    float  $pdfConValue, docConValue$            ▷ BFAs confidence values
    byte[] $byteStream$                          ▷ Byte content of fragment

Ensure: $XLS, PDF, DOC, TEXT$                    ▷ Classification result

 1: declare integer $ninb$        ▷ Number of Individual Null Bytes in fragment
 2: declare float $entropy$                      ▷ Entropy of the fragment
 3: declare float $ptc$               ▷ Plain Text Concentration in fragment

 4: declare const integer $lowNinb := 9$
 5: declare const integer $highNinb := 25$
 6: declare const float $textMaxEntropy := 6$
 7: declare const float $xlsMaxPtc := 50$
 8: declare const integer $xlsMinNinb := 50$
 9: declare const float $medianPdfEntropy := 5.8$
10: declare const float $lowEntropy := 3.9$
```

Auxiliary functions (Part II)

```
**function** isXls()
    **return** $ninb > xlsMinNinb \ \wedge \ ptc < xlsMaxPtc$
**end function**

**function** isPdf()
    **return** $pdfConValue > docConValue \ \wedge \ ninb \leq lowNinb \vee$
            $entropy \geq medianPdfEntropy$
**end function**

**function** isNotPlainText()
    **return** $ptc \neq 100$
**end function**

**function** isNotPdf()
    **return** $entropy \leq lowEntropy \ \wedge \ ninb \geq highNinb$
**end function**
```

Classifier Part(III)

```
if isNotPlainText( ) then
    if isXls( ) then
        return $XLS$
    else if isNotPdf( ) then
        return $DOC$
    else if isPdf( ) then
        return $PDF$
    else
        return $DOC$
    end if
else
    if $entropy < textMaxEntropy$ then
        return $TEXT$
    else if isPdf( ) then
        return $PDF$
    else
        return $DOC$
    end if
end if
```