Aufgabe 1 Lösung

a)

Vereinfachte BNF

Vereinfachte BNF aus der Java Spezifikation, ohne binäre und hexadezimale Schreibweisen und ohne Suffixe.

```
FloatingPointLiteral: DigitPart ExponentPart_opt
DigitPart:            Digits . Digits_opt
                    | . Digits
                    | Digits
ExponentPart:         ExponentIndicator SignedInteger
ExponentIndicator:    e | E
SignedInteger:        Sign_opt Digits
Sign:                 + | -
Digits:               Digit | Digits Digit
Digit:                one of 0 1 2 3 4 5 6 7 8 9
```

Regulärer Ausdruck

Schritt 1: Definiere reguläre Ausdrücke für Teilprobleme

Schritt 2: Setze Teilausdrücke zusammen

Schritt 3: Falte Teilausdrücke aus und vereinfache

Schritt 1 und 2 Einen vorher definierten regulären Ausdrück RE verwende ich später mit der Syntax `{RE}` innerhalb anderer REs.

```
RE_digit = [0-9]
RE_digits = {RE_digit}+
RE_signedInteger = [-+]?{RE_digits}
RE_exponentPart = [eE]{RE_signedInteger}
RE_digitPart1 = {RE_digits}.({RE_digits})?
RE_digitPart2 = .{RE_digits}
RE_digitPart3 = {RE_digits}
RE_digitPart = ({RE_digit1})|({RE_digit2})|({RE_digit3})
RE_float = ({RE_digitPart})({RE_exponentPart})?
```

Schritt 3 Ausgefaltet und vereinfacht ergibt das folgendenen RE:

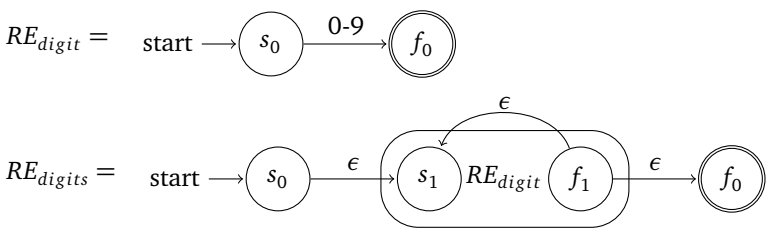
```
(([0-9]+.[0-9]*)|(.[0-9]+)|([0-9]+))([eE][-+]?[0-9]+)?
```

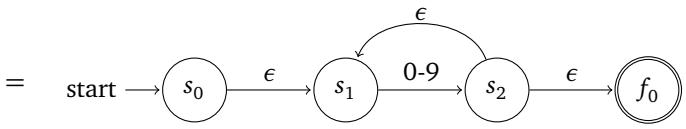
Kann noch weiter vereinfacht werden zu:

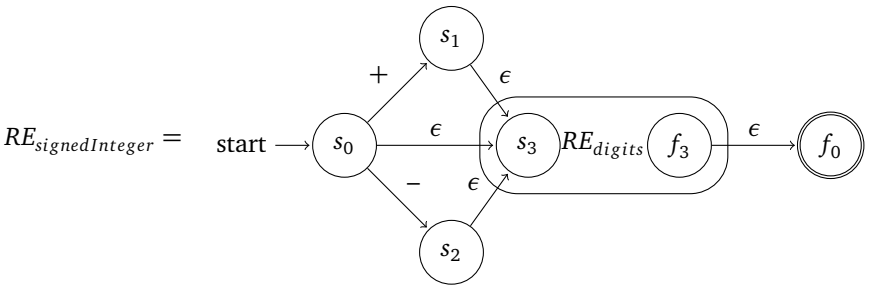
```
(([0-9]+(.[0-9]*)?)|(.[0-9]+))([eE][-+]?[0-9]+)?
```

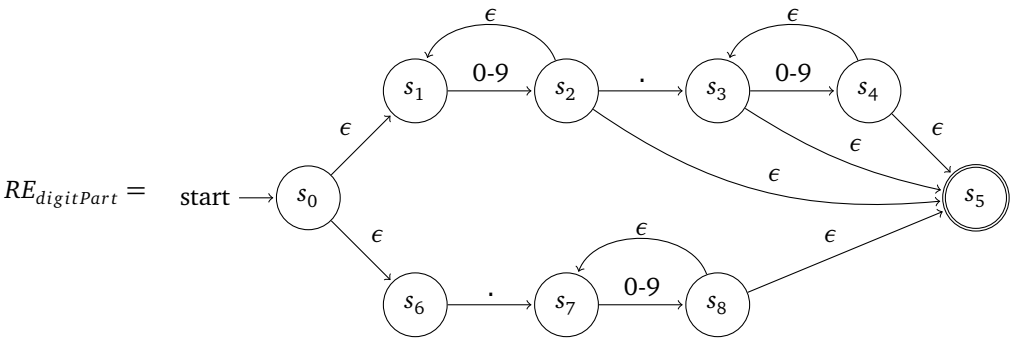
b)

NFAs für Teilausdrücke



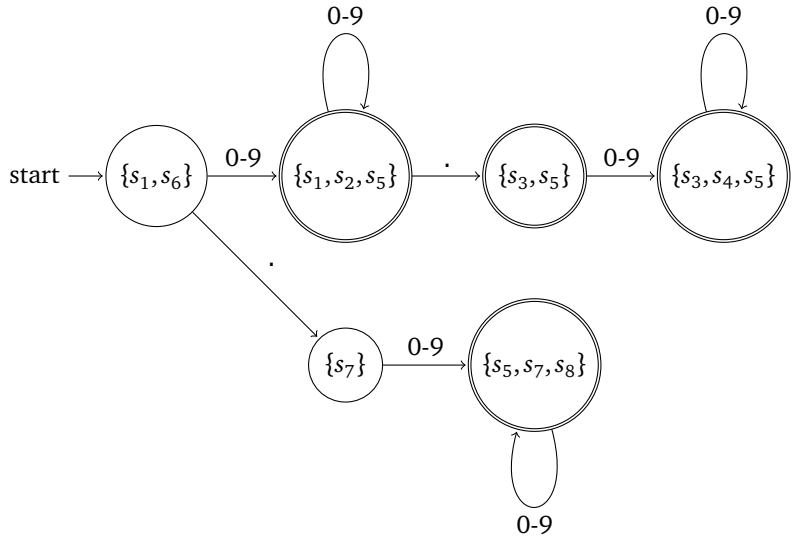

=
start
s_0
ϵ
s_1
0-9
s_2
ϵ
f_0
ϵ


$RE_{signedInteger}$ =
start
s_0
+
s_1
ϵ
ϵ
s_3
RE_{digits}
f_3
ϵ
f_0
−
ϵ
s_2


$RE_{digitPart}$ =
start
s_0
ϵ
s_1
0-9
s_2
.
s_3
0-9
s_4
ϵ
ϵ
ϵ
ϵ
s_5
ϵ
s_6
.
s_7
ϵ
0-9
s_8
ϵ

c)

DFA


start
$\{s_1, s_6\}$
0-9
$\{s_1, s_2, s_5\}$
0-9
.
$\{s_3, s_5\}$
0-9
$\{s_3, s_4, s_5\}$
.
$\{s_7\}$
0-9
$\{s_5, s_7, s_8\}$
0-9

Simplified DFA