

Aufg. 1.3

Java Gleitkommazahlen Bedingungen:

- Trenner ist ein Punkt .
- min 1Zahl vor oder nach dem Punkt
- beliebig viele Zahlen vor oder nach dem Punkt
- zum Schluss kommt ein Buchstabe f F oder d D, optional

Möglichkeit Exponent:

- Exponent genau ein Buchstabe e oder E
- nach e kann + oder - oder Zahl stehen
- nach e eine oder beliebig viele Zahlen
- vor e kann eine Zahl stehen

Grob:

- ⇒ [keine/ beliebig viele Zahlen][.][keine/ beliebig viele Zahlen][ein oder kein Buschstabe]
- ⇒ Aber wenn vor Komma, dahinter muss keine Zahl sein
Wenn hinter komme, vorne muss keine Zahl sein

Exponent:

- ⇒ ([keine/ beliebig viele Zahlen][Buchstabe e][kein oder ein Vorzeichen][mind. 1 Zahl])?
Der gesamte Exponenten-Ausdruck steht entweder einmal oder keinmal
Die vorstehende Zahl kommt aus dem anderen Ausdruck

Zusammengefasst:

([0-9]* \.[0-9]+)|([0-9]+)\.([0-9]*)[fFdD]?

Das ‚oder‘ verhindert, dass ein Punkt allein akzeptiert wird.

keine/beliebige Zahlen ‚Punkt‘ mindestens 1 Zahl oder

mindestens 1 Zahl ‚Punkt‘ keine/beliebige Zahl

Exponent: ([eE][+-]?[0-9]+)?

Zusammengesetzt:

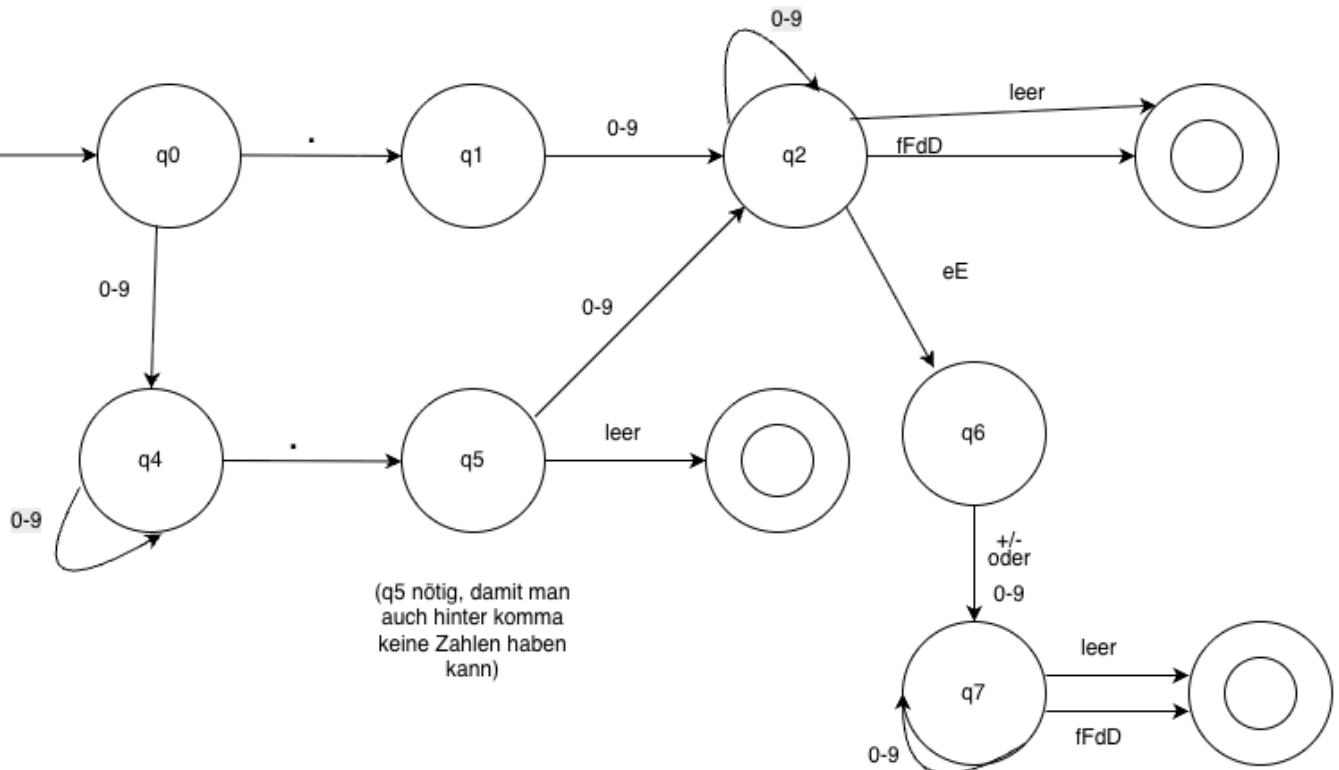
(([0-9]* \.[0-9]+)|([0-9]+)\.([0-9]*))([eE][+-]?[0-9]+)? [fFdD]?

Der obige Ausdruck, gefolgt von einem oder keinem Exponenten

Bei Python gibt es kein zum Schluss stehendes d/f, der Rest gilt. Der Ausdruck ist also:

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

DFA-Versuch:



Der Graph gilt für Java. Für Python müssten nur die beiden Pfeile mit fFdD wegfallen.

Ich bin noch unsicher, wann man einen neuen Endzustand machen sollte bzw. darf.

Beispiel .43E-12d

Start bei q0. Eingabe . -> geht zu q1.

q1 akzeptiert nur Zahlen. Ein e würde nicht akzeptiert. Eingabe:4, 3 bleibt in Zustand q2.

Eingabe E, wechselt zu q6.

Eingabe -: Wechselt zu q7

Eingaben 1, 2: bleibt in q7

Eingabe d: wechselt in Endzustand.

Beispiel: 124.89

Eingaben 1: Wechselt in q4.

Eingaben 2, 4: bleibt in q4

Eingabe . : wechselt nach q5

Eingabe 8 (9): wechselt, bleibt in q2.

Eingabe leer: wechselt in Endzustand.

Grammatiken:

Q0-> . Q1 | 0-9 Q4

Q1-> 0-9 Q2

Q2-> 0-9 Q2 | eE Q6 | leer END | fFdD END

(Q3 habe ich beim Nummerieren übersehen)

Q4-> 0-9 Q4 | . Q5

Q5-> 0-9 Q2 | leer END

Q6 -> +- oder 0-9 Q7

Q7 ->0-9 Q7 | leer END | fFdD END

Für Python ohne fFdD

Q0-> . Q1 | 0-9 Q4

Q1-> 0-9 Q2

Q2-> 0-9 Q2 | eE Q6 | leer END

(Q3 habe ich beim Nummerieren übersehen)

Q4-> 0-9 Q4 | . Q5

Q5-> 0-9 Q2 | leer END

Q6 -> +- oder 0-9 Q7

Q7 ->0-9 Q7 | leer END

Bäume:

.43E-12d

