**C2 Syntaxprüfer**

Wenn bei der lexikalischen Analyse eines Javaprogramms durch den Compiler der Programmtext in die Sinneinheiten der Programmiersprache, die *tokens*, zerlegt ist, muss jede solche Zeichenfolge daraufhin geprüft werden, ob sie ein Wortsymbol oder einen Bezeichner darstellt oder keins von beiden. Wir machen uns dies beispielhaft klar, indem wir einen erkennenden Automaten konstruieren, der die Wortsymbole „do“ und „downto“ sowie Bezeichner unterscheidet. Wir nehmen dabei vereinfachend an, dass diese beiden Wortsymbole die einzigen sind.

Beispiel: Syntaxprüfer für überlappende Begriffe:

Der Automat (X, Z, δ, z0, ZE) habe das Eingabealphabet X = A ∪ B ∪ C, wobei A die Menge der Sonderzeichen, B die Menge der (Klein-)Buchstaben und C die Menge der Ziffern darstellt. Seine Zustandsmenge sei Z = {z0, z1, …, z4} und seine Endzustandsmenge ZE = {z2, z4}. Die Überführungsfunktion ist durch den Zustandsgraphen gegeben. Der Automat erkennt im Endzustand z2 das Wortsymbol „do“ und im Endzustand z4 einen Bezeichner.

Z3

Z4

Z2

Z1

Z0

A ∪ C

d

o

A

A ∪ B ∪ C

B \ {d}

A

B ∪ C \ {o}

B ∪ C

B ∪ C

A

Aufgaben:

1. Gib drei mögliche Eingaben an, die von dem oben abgebildeten Automat akzeptiert werden.
2. Welches Wortsymbol wird durch diesen Automat erkannt?
3. Woran erkennt der Automat einen Bezeichner?
4. Akzeptiert der Automat auch das Wort „downtown“? Begründe!
5. Ergänze den Graphen so, dass mit möglichst wenig weiteren Zuständen zusätzlich auch das Wortsymbol „downto“ erkannt wird.