

# **Formale Sprachen**

# Überblick

Das Fachgebiet „Formale Sprachen“ der Informatik beschäftigt sich u.a. mit

- der Struktur von **Programmiersprachen**
- der Entwicklung von **Compilern**

Automatenmodelle werden auch verwendet, um Algorithmen auf **Zeitbedarf** und **Berechenbarkeit** zu untersuchen.

# Compiler

**1. Prüft** Quelltext in einer „höheren“ Programmiersprache (z.B. Java) auf **syntaktische** Korrektheit.

**2. Übersetzt** Quelltext in Maschinencode.

Syntax = Grammatik

Semantik = Bedeutung

(hier: Funktionsweise des Programms)

Die Semantik eines Programms kann der Compiler nicht prüfen.

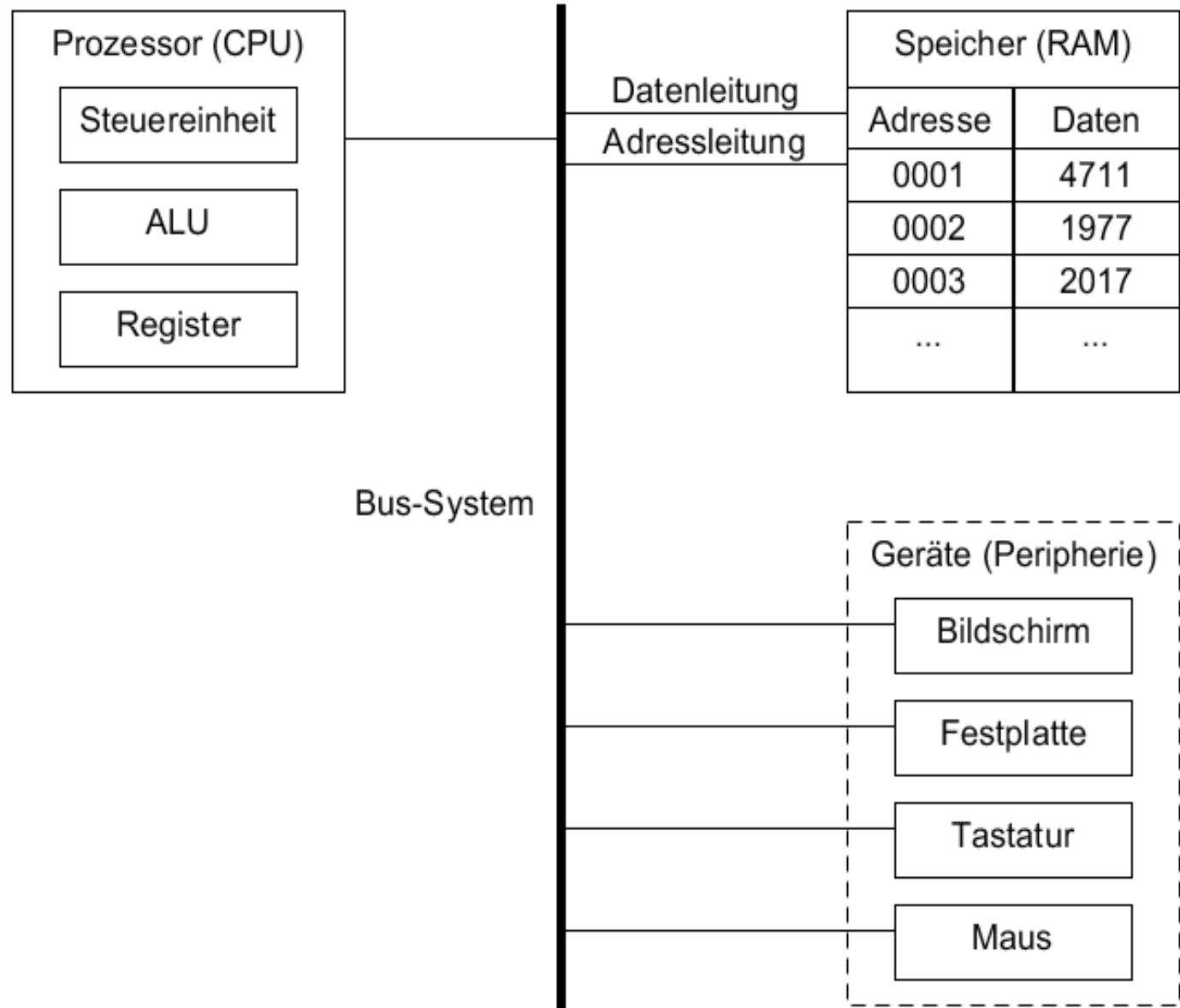
# Von-Neumann-Rechner

## CPU-Befehle

- Daten in RAM speichern
- Daten aus RAM lesen
- Daten an Gerät senden
- Daten von Gerät lesen
- Rechnen mit Daten
- Daten vergleichen
- Sprung (für Schleifen)

Befehle sind codiert als Zahlen.  
Ein Programm besteht aus diesen Codes und liegt im Speicher. D.h. die CPU holt sich ihre Befehle aus dem RAM.

Besonderheit der  
**von-Neumann-Architektur:**  
Programme und Daten liegen im gleichen Speicher.



# Modell: „Endlicher Automat“

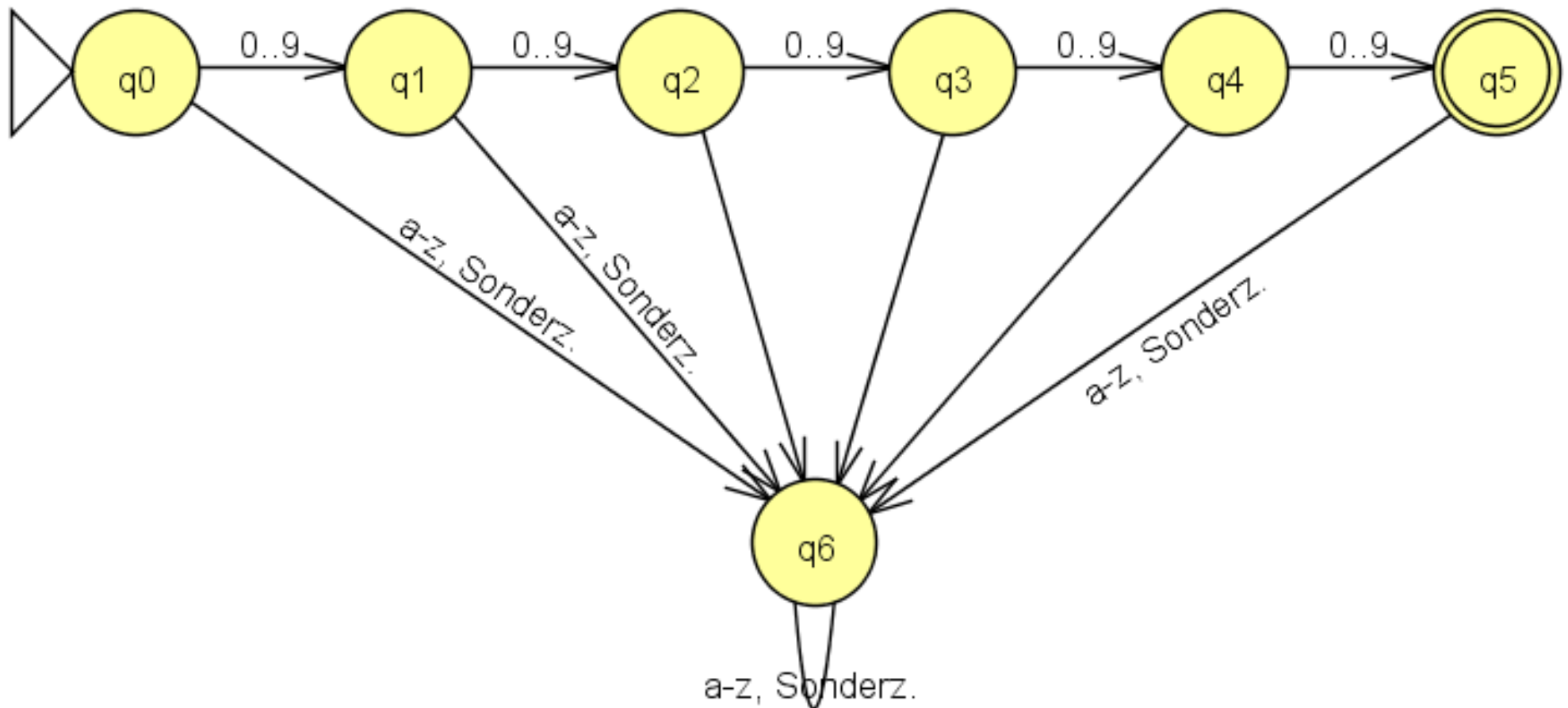
Prüft, ob ein Text einer bestimmte Form entspricht (wesentlich einfacher als die Prüfung eines Java-Programms)

Beispiele: Prüfe, ob die Eingabe ...

- eine gültige Emailadresse
- eine gültige Postleitzahl
- ein gültiges Datum (etc.) ... ist
- ein bestimmtes Wort enthält

# Beispiel: Postleitzahl

Zu prüfen: Hat eine Eingabe das Format einer Postleitzahl?

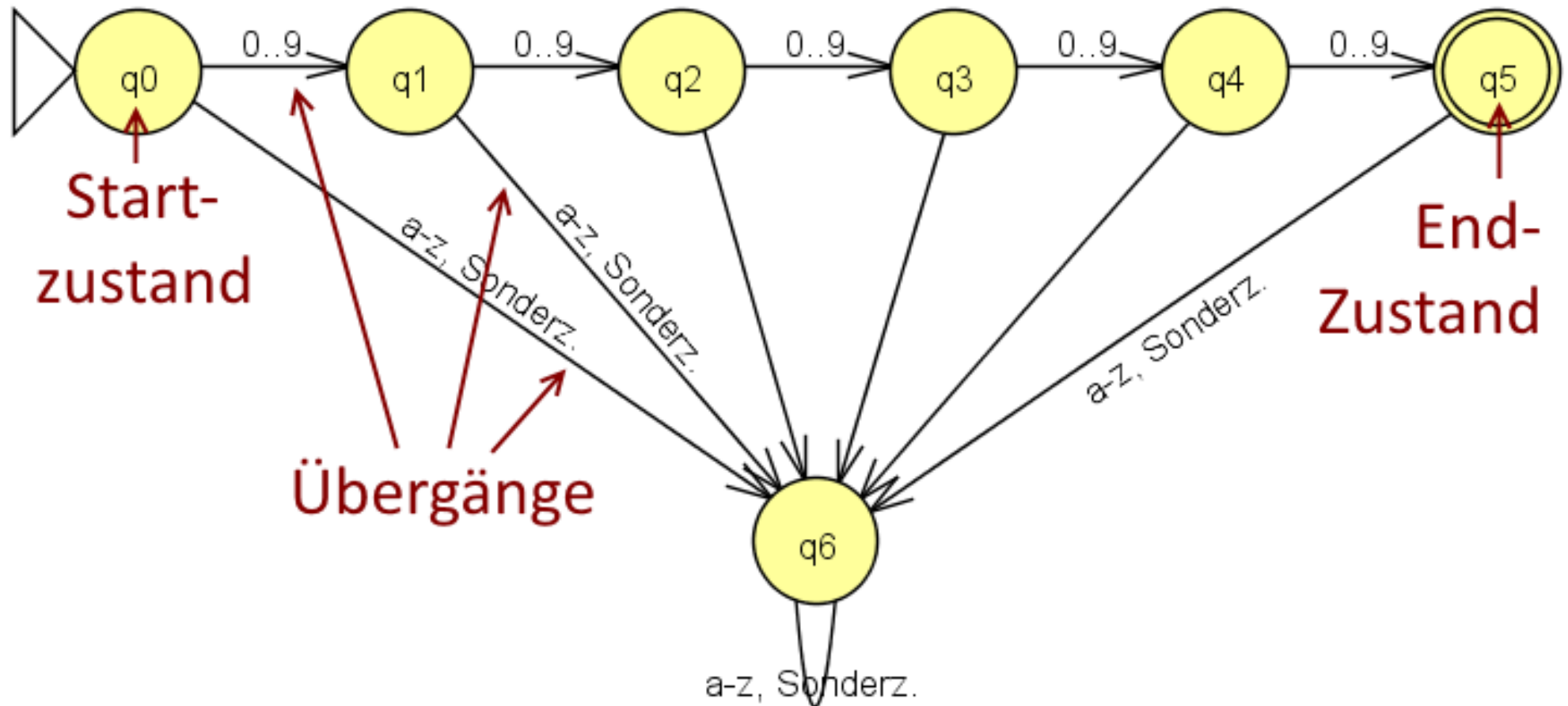


# Endlicher Automat

Ein endlicher Automat hat **Zustände**  
(dargestellt durch Kreise)  
sowie **Zustandsübergänge** (Pfeile).  
Es gibt einen **Startzustand**,  
sowie einen oder mehrere **Endzustände**.

Ein endl. Automat prüft **Eingabewörter**  
(d.h. beliebige Zeichenketten)  
über einem **Alphabet**  
(d.h. eine beliebige Menge von Zeichen).

# Beispiel: Postleitzahl

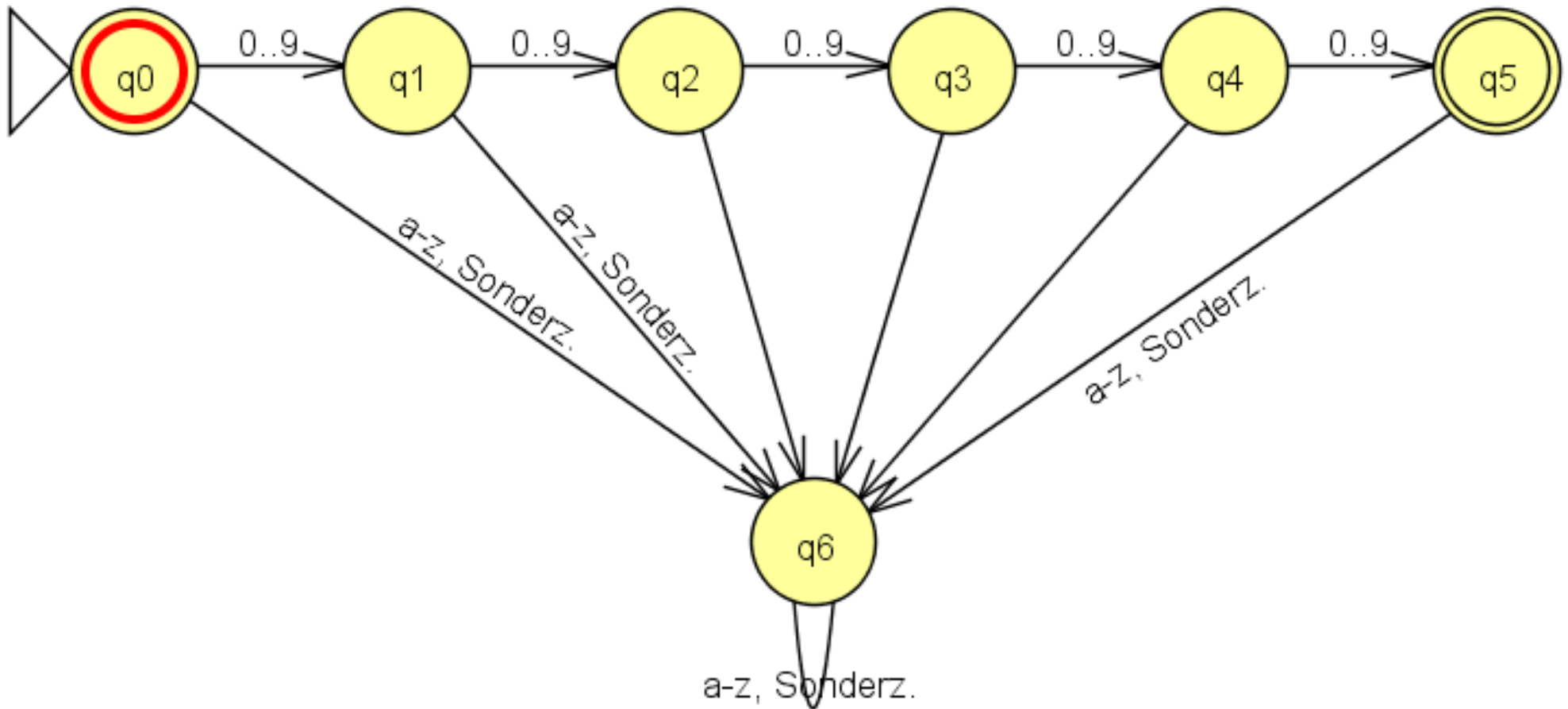




# Beispiel: Postleitzahl

Prüfe die Eingabe: 50823

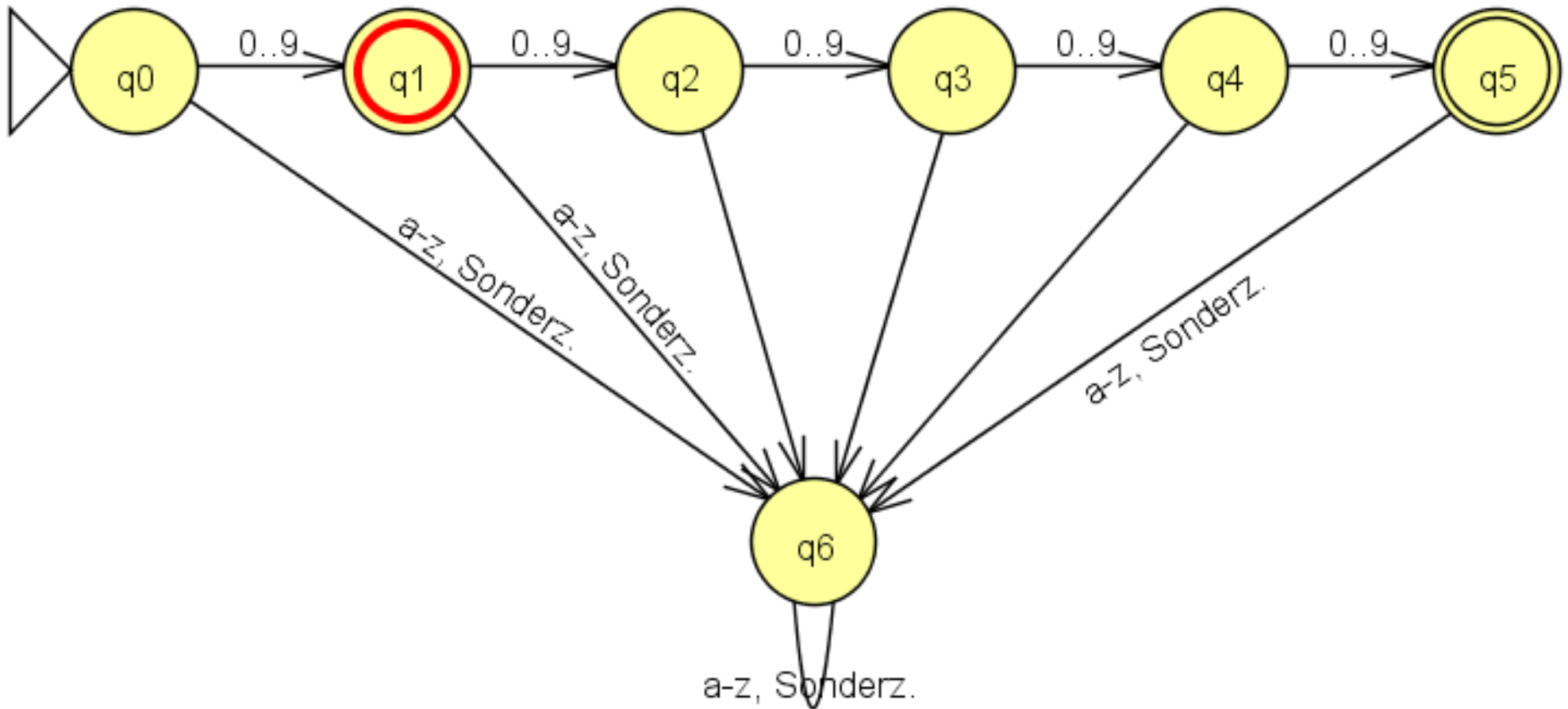
Automat beginnt im Startzustand



# Beispiel: Postleitzahl

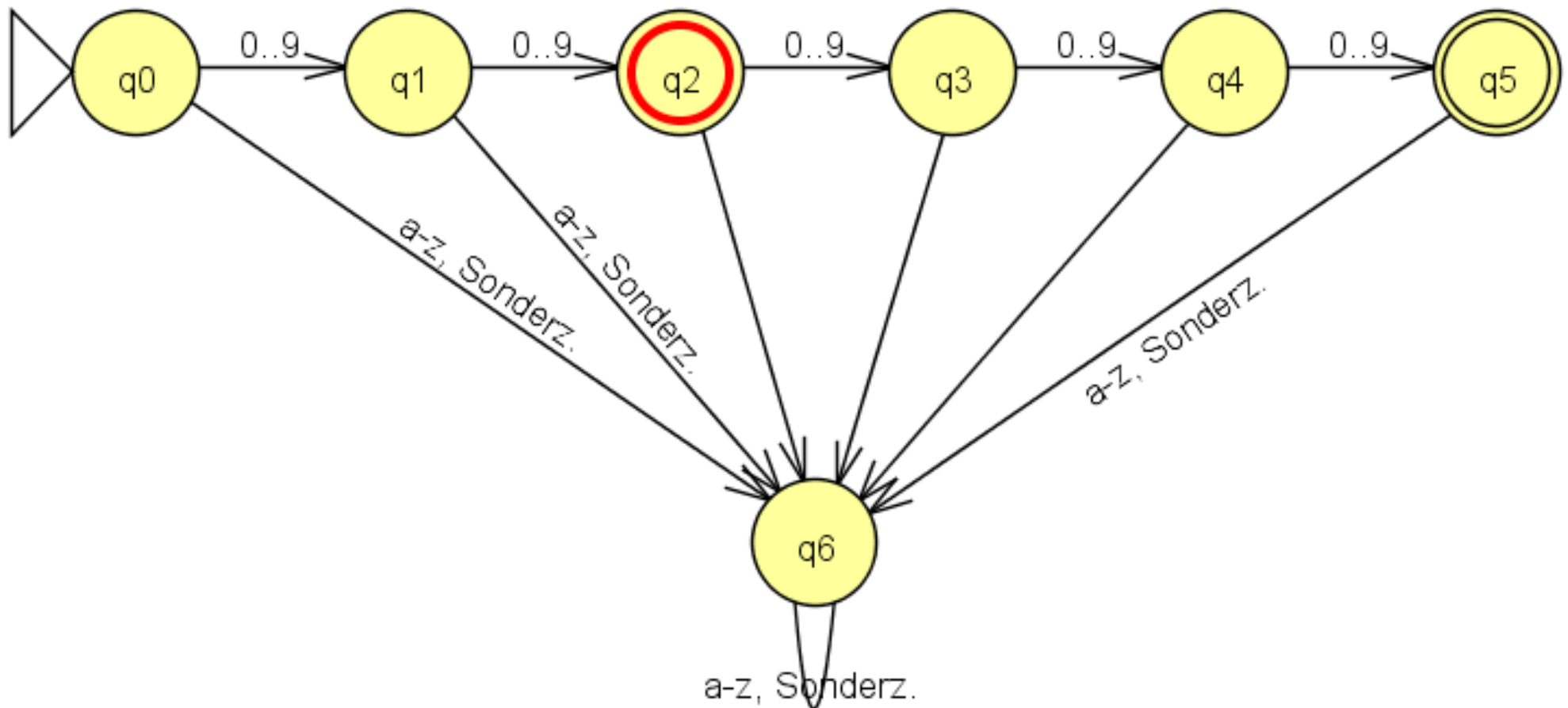
Prüfe die Eingabe: 50823

Für jedes Zeichen ein Übergang



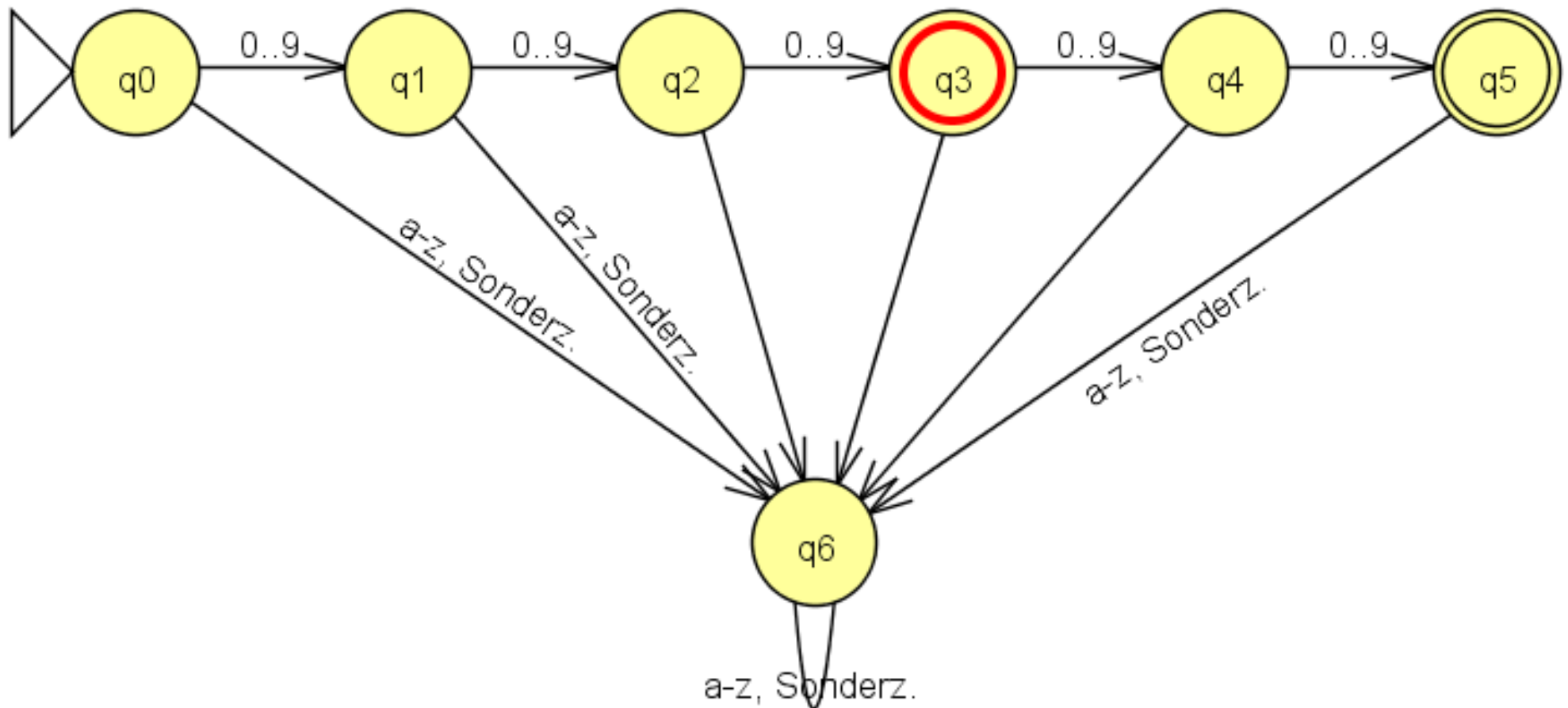
# Beispiel: Postleitzahl

Prüfe die Eingabe: 50823



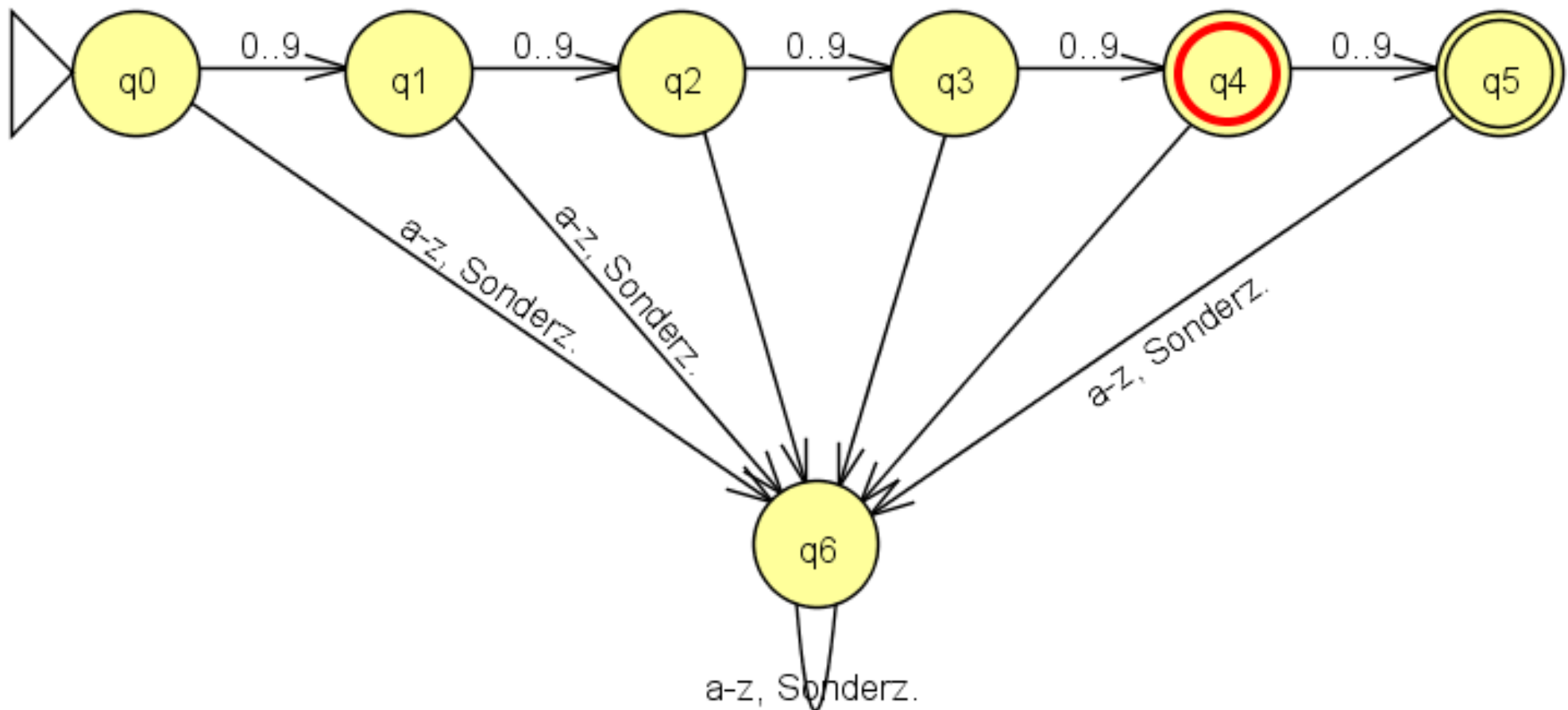
# Beispiel: Postleitzahl

Prüfe die Eingabe: 50823



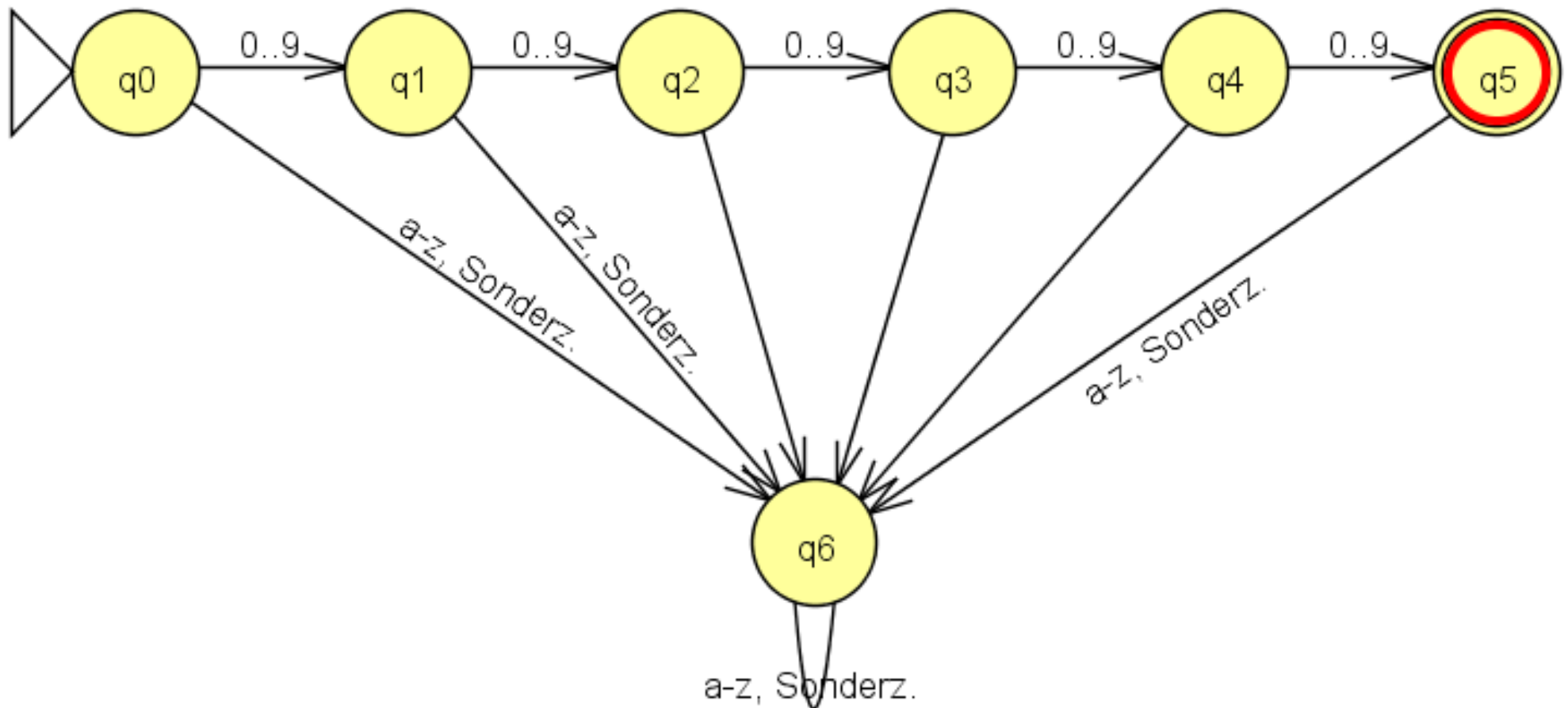
# Beispiel: Postleitzahl

Prüfe die Eingabe: 50823



# Beispiel: Postleitzahl

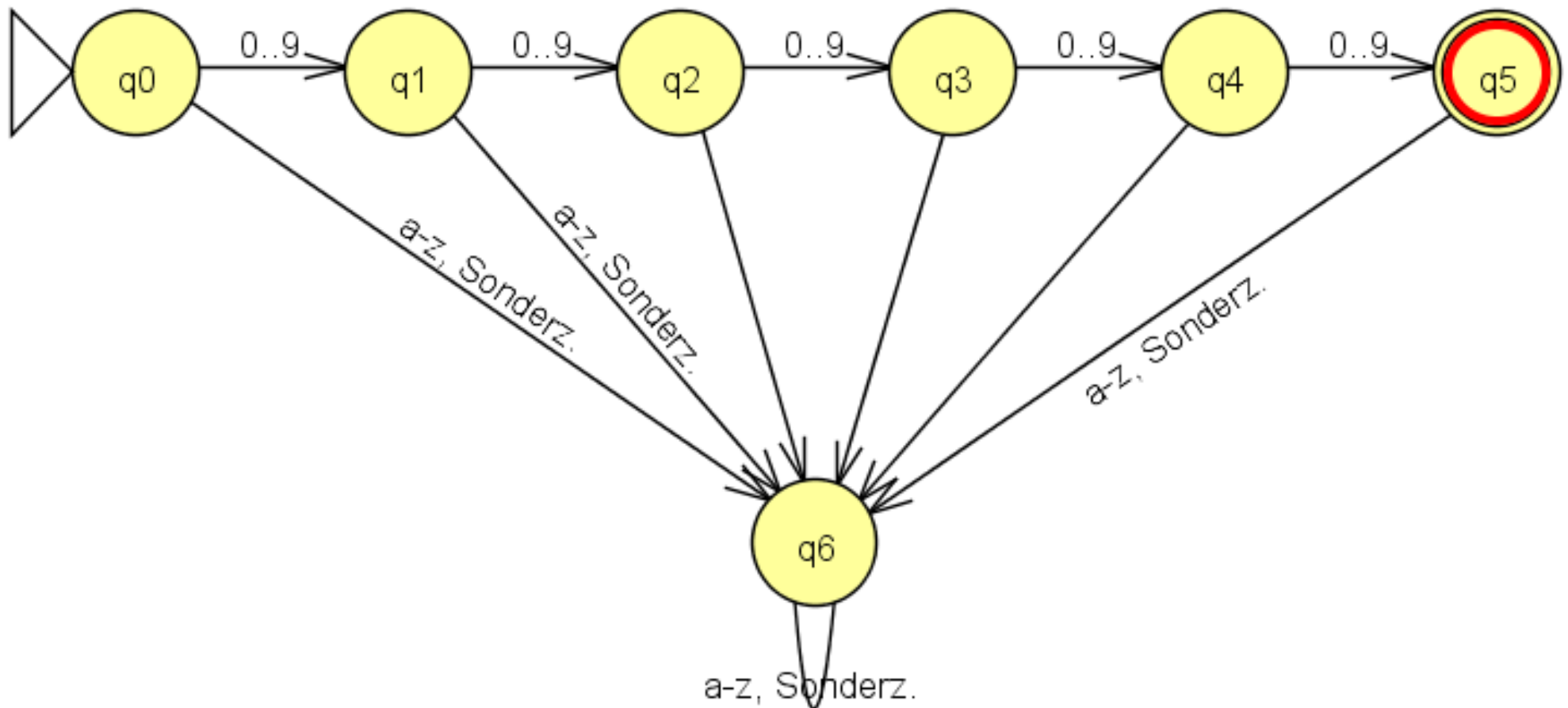
Prüfe die Eingabe: 50823



# Beispiel: Postleitzahl

Die Eingabe 50823 wurde bis zum Ende gelesen, ein Endzustand ist erreicht.

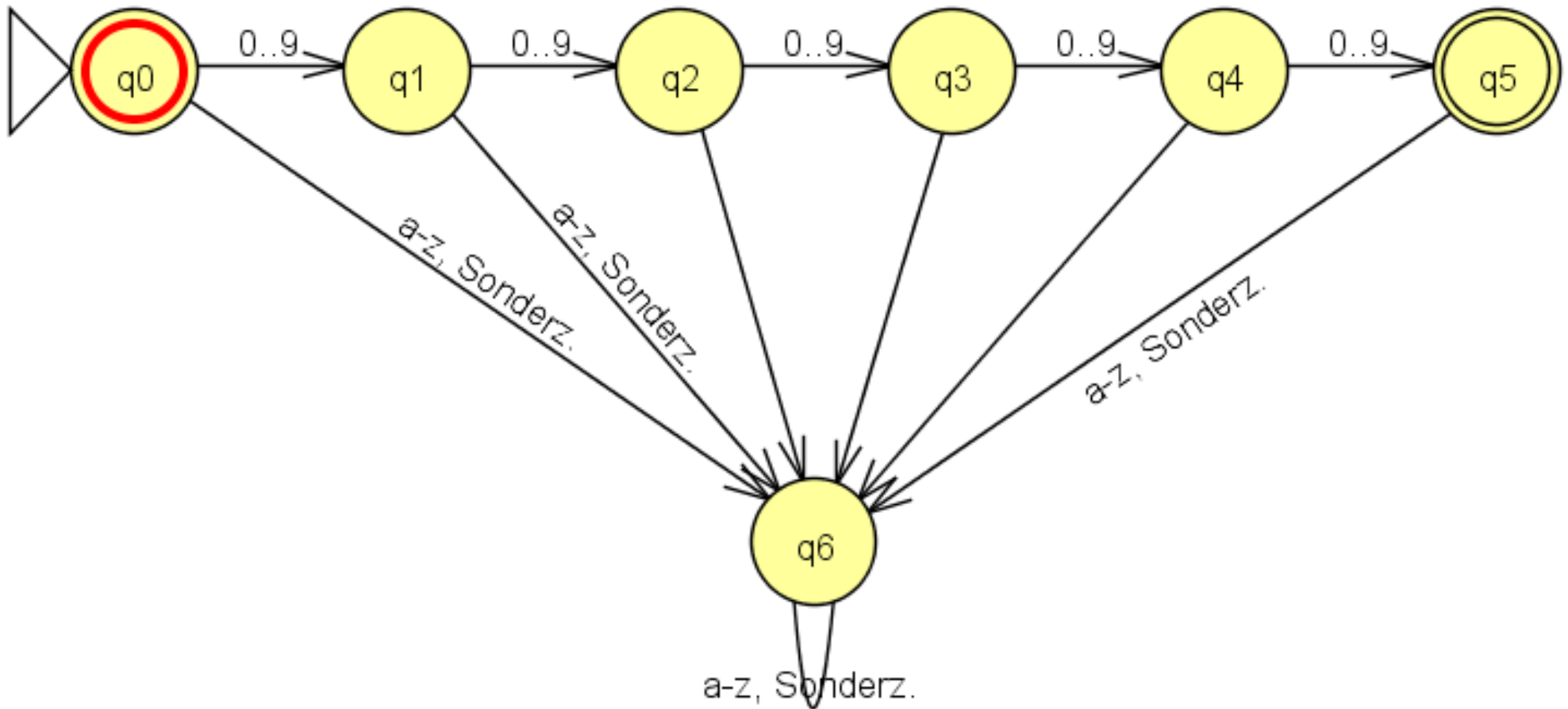
→ Die Eingabe wird **akzeptiert**.



# Beispiel: Postleitzahl

Prüfe die Eingabe: 5A12C

Automat beginnt im Startzustand

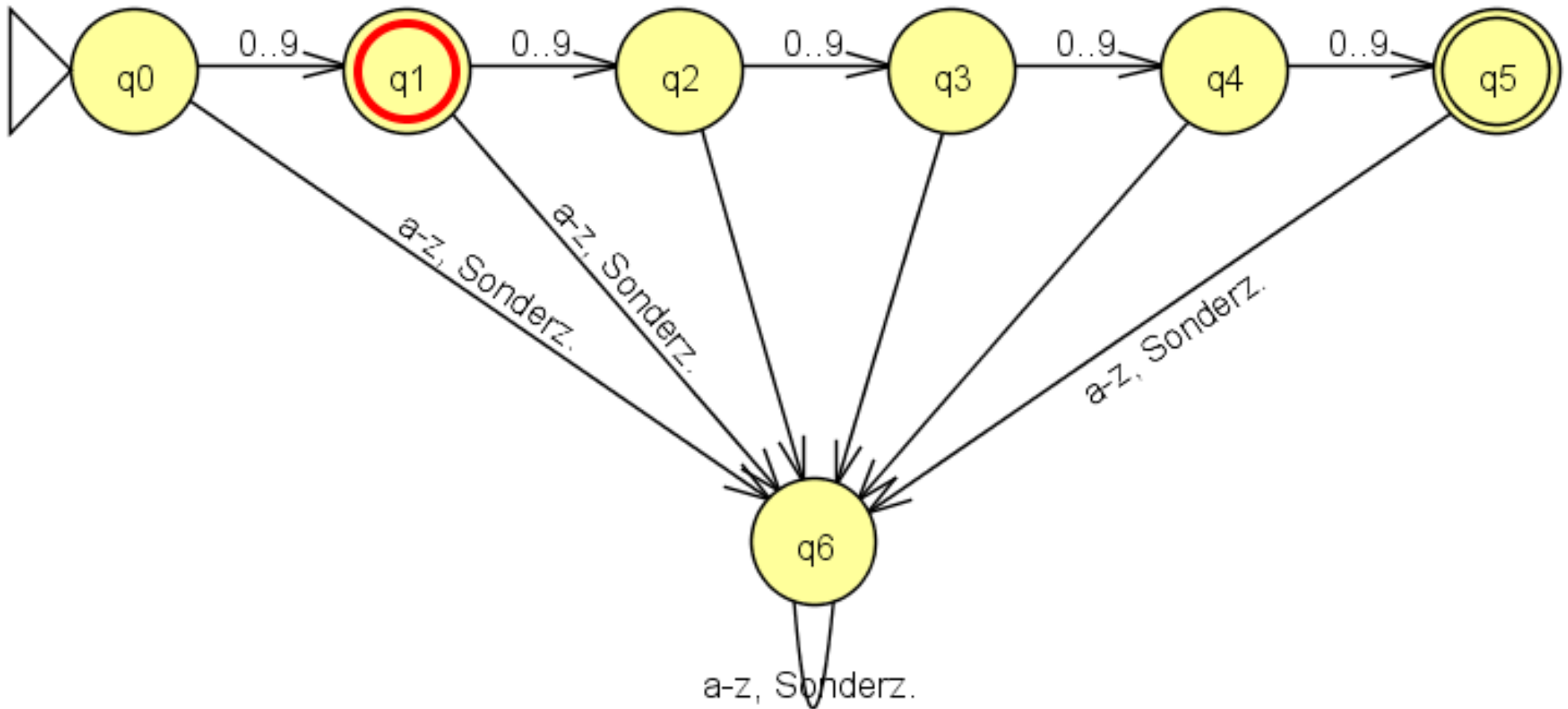




# Beispiel: Postleitzahl

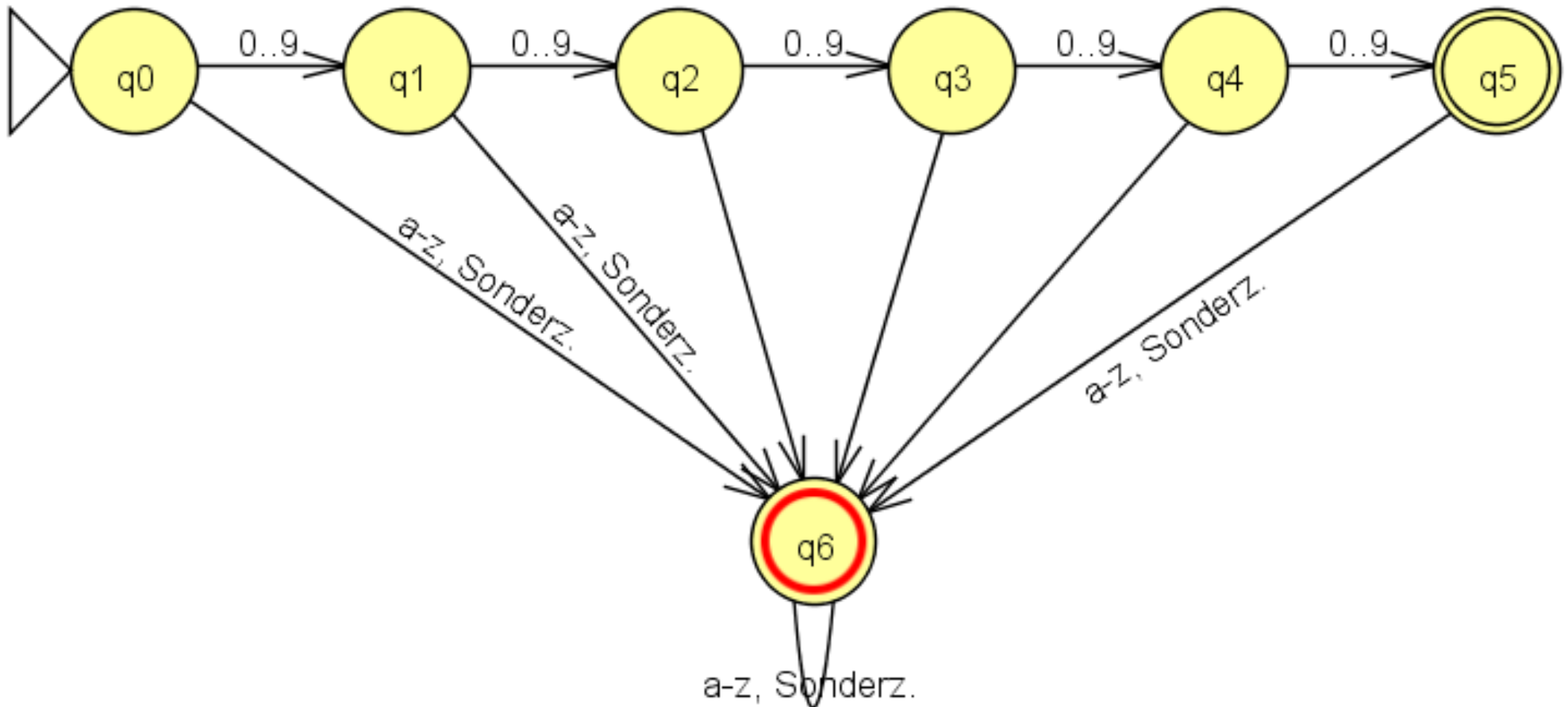
Prüfe die Eingabe: 5A12C

Für jedes Zeichen ein Übergang



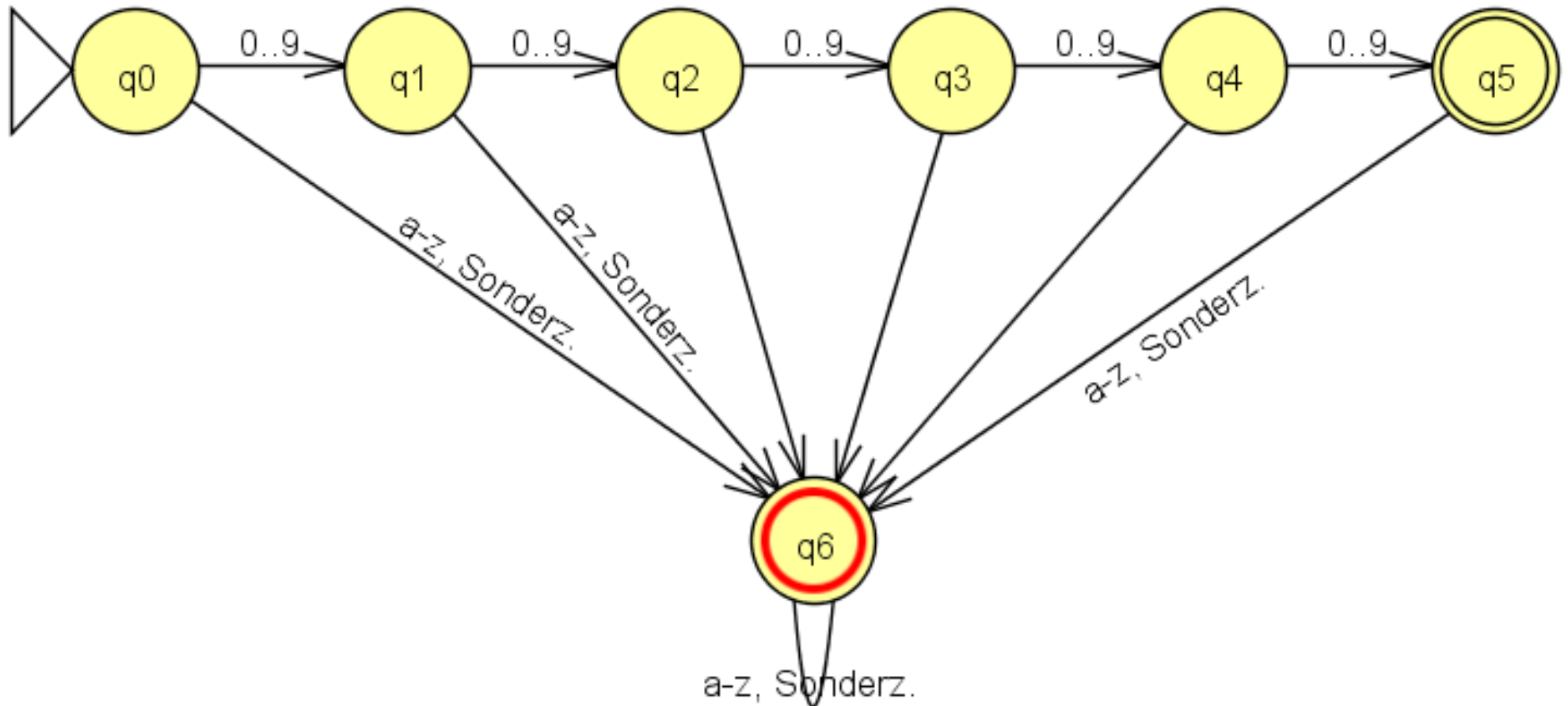
# Beispiel: Postleitzahl

Prüfe die Eingabe: 5A12C



# Beispiel: Postleitzahl

Den Zustand q6 kann der EA nicht verlassen. Da q6 kein Endzustand ist, wird die Eingabe 5A12C **nicht akzeptiert**.



# Aufgabe 1

Entwirf einen endl. Automaten, der prüft, ob die Eingabe ein korrektes **Datum** ist.

- Eingaben: nur Ziffern und Punkte (keine Buchstaben oder sonst. Zeichen)
- Tage 1 – 31, Monate 1 – 12, Jahre 0 – 2999
- Keine Prüfung, ob z.B. 30.02. oder 31.04. korrekt ist

# Aufgabe 2

Entwirf einen endl. Automaten, der prüft, ob die Eingabe eine **Straßenadresse** ist.

- Beispiel: „Körnerstr. 50“
- Überlege dir zunächst weitere Beispiele.
- Notiere, nach welchen Regeln Straßenadressen aufgebaut sind.
- Entwirf dann den endl. Automaten.
- Eingaben:  
Groß- und Kleinbuchst., Zeichen, Ziffern

# Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)  
Freigegeben unter CC BY-NC-SA 4.0, März 2022

