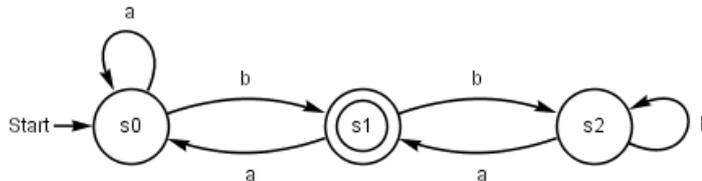


Erkennender, endlicher Automat als geeignetes Werkzeug zur Syntaxprüfung für reguläre Sprachen; Implementierung eines endlichen Automaten

- prüft syntaktische Korrektheit der Eingaben (z.B. E-Mail-Adresse, Passwort, KFZ- Kennzeichen)
- nach Überprüfung wird ein entsprechender Rückgabewert ausgegeben (true/ false)
- Darstellung im **Zustandsübergangsdiagramm**



- Startzustand: „s0“
- Eingabe muss mindestens ein „b“ enthalten → Endzustand „s1“ wird erst erreicht, wenn einmal „b“ eingegeben wurde
- Eingabe kann beliebig viele Zeichen besitzen
- die letzten zwei Zeichen dürfen nicht gleich sein
- das Diagramm kann durch einen „Fangzustand“ erweitert werden → Zustand, der nicht mehr verlassen werden kann → kann durch eine ungültige Eingabe erreicht werden (im Bsp. wäre „c“ eine ungültige Eingabe)
- **Implementierung in Java:**
 - Start-Methode → Boolean-Methode
 - Startzustand wird festgelegt
 - Es wird immer nur der erste Buchstabe des Strings betrachtet, solange der String nicht leer ist (`while (s.length() > 0)`)
 - Folge von **if-Schleifen:**
 - ```
if ((zustand.equals("s0")) && (ista(s.substring(0,1)))) zustand="s0"; else
 if ((zustand.equals("s0")) && (istb(s.substring(0,1)))) zustand="s1"; else
...

```

→ diese werden für alle Möglichkeiten für alle Zustände festgelegt
  - erneuter Durchgang mit nächstem Buchstaben (`s = s.substring(1)` → String ab zweitem Zeichen ist neuer String s)
  - wurden alle Zeichen des eingegebenen Strings überprüft, gibt die Methode einen entsprechenden Rückgabewert aus (wenn aktueller Zustand Endzustand ist → „true“; ansonsten „false“)
  - Eingabe gültig bzw. ungültig