5.11 Reguläre Ausdrücke in der Praxis

ZUSATZ

> ZUSATZ
>
> Reguläre Ausdrücke stehen in keinem der beiden Lehrpläne. Sie sind hier aufgenommen, weil sie dieselbe Sprachklasse beschreiben wie Grammatik und Automat — und weil sie das einzige Werkzeug dieses Kapitels sind, das Ihnen im Alltag jede Woche begegnet.

Sie wollen in einem Chatverlauf alle Nachrichten finden, in denen eine Uhrzeit steht, egal ob 7:30 oder 18:05. Die normale Suche verlangt jede Uhrzeit einzeln. Ein regulärer Ausdruck beschreibt stattdessen das *Muster* einmal: `[0-9]+:[0-9][0-9]` — eine oder mehr Ziffern, ein Doppelpunkt, zwei Ziffern. Er benutzt nur Bausteine aus der Tabelle dieses Abschnitts.

Was in 5.5 als Grammatik und in 5.6 als Automat auftrat, schreibt man im Alltag als **regulären Ausdruck**. Editoren, Suchfunktionen, Formularprüfungen und Logauswertungen benutzen sie täglich.

Ausdruck	Bedeutung	passt auf
`abb`	genau diese Folge	`abb`
`a*`	beliebig viele a, auch keines	`ε`, `a`, `aaa`
`a+`	mindestens ein a	`a`, `aaa`
`a?`	höchstens ein a	`ε`, `a`
`[a-z]`	ein Zeichen aus dem Bereich	`k`
`(ab)+`	Wiederholung der Gruppe	`ab`, `ababab`
`a|b`	Alternative	`a` oder `b`

Ein Bezeichner ist damit `[A-Za-z][A-Za-z0-9]*`, eine Dualzahl ohne führende Null `1[01]*`, eine gerade Dualzahl `1[01]*0`.

```python
import re

muster = re.compile(r"[A-Za-z][A-Za-z0-9]*$")
for text in ["zaehler", "2fast", "x1", "mein wert"]:
    print(text, bool(muster.match(text)))
```

Unter der Haube übersetzt die Bibliothek den Ausdruck in einen endlichen Automaten — und läuft ihn genau so ab, wie Sie es in 5.9 programmiert haben.

Arbeitsauftrag 11

44. Schreiben Sie reguläre Ausdrücke für: Wörter über $\{a, b\}$ mit `aa`; ungerade Dualzahlen; deutsche Postleitzahlen; eine E-Mail-Adresse in grober Näherung.
45. Prüfen Sie mit dem Python-Beispiel Ihre Ausdrücke aus Aufgabe 44.
46. Begründen Sie, warum sich mit regulären Ausdrücken nicht prüfen lässt, ob in einem Term gleich viele öffnende wie schließende Klammern stehen.