LEHRERINFO ZUM VIDEO

Automaten und Grenzen der Berechenbarkeit

FACH	STUFE	DAUER	VIDEO
Informatik	Gymnasium · Sek II	12 min	youtube.com/@Erklaerbaer

01 Lernziele

Nach dem Video können die Schülerinnen und Schüler …

- endliche Automaten als Zustandsdiagramm lesen und zeichnen (Zustände, Übergänge, Startzustand, akzeptierende Zustände),
- mit einem Automaten Schritt für Schritt prüfen, ob eine Eingabe akzeptiert wird (Teilbarkeit durch 3),
- einfache reguläre Ausdrücke lesen und schreiben (`[0-9]{4}`, `^ $ + * ?`),
- begründen, warum endliche Automaten beliebig tief geschachtelte Klammern nicht prüfen können, und den Kellerautomaten als Erweiterung beschreiben,
- Aufbau und Arbeitsweise einer Turingmaschine erklären und ein kleines Programm von Hand ausführen,
- den Widerspruchsbeweis zum Halteproblem nachvollziehen und Folgen wie die Grenzen von Virenscannern einordnen.

02 Voraussetzungen

Bedingungen und Schleifen in Python (Folge 8 · Grundlagen); Terminierung von Algorithmen (Folge 6); Binärzahlen (Folge 2).

03 Ablauf des Videos

ZEIT	KAPITEL	INHALT UND BEISPIEL
0:00	Einstieg	Kann ein Programm vorhersagen, ob ein anderes anhält? · Turing: unmöglich · Ident · Fahrplan
0:36	Was du brauchst	Bedingungen, Schleifen, Terminierung, Binärzahlen · Hinweis auf Folge 8 Grundlagen

0:54	Zustände statt Befehle	Drehkreuz (gesperrt/offen) · Zustandsdiagramm · Getränkeautomat 1,50 € mit Rückgeld · Merksatz
2:11	Automaten, die Eingaben prüfen	akzeptierende Zustände · Teilbarkeit durch 3: 110 PAUSE-FRAGE 1001 · Python-Schleife · Spielgegner
3:43	Reguläre Ausdrücke	`[0-9]{4}` als Zustandskette · PLZ 8003 · `^ $ + * ?` · Formular, Editor PAUSE-FRAGE Kennzeichen ZH 123456
5:01	Wo endliche Automaten scheitern	Klammern zählen · Kellerautomat mit Stapel · Leiter der Maschinen
5:42	Die Turingmaschine	Band, Kopf, Tabelle · 1011 + 1 = 1100 · Church-Turing-These · Merksatz
6:45	Das Halteproblem	Halte-Tester und Trotzkopf · Widerspruch · Virenscanner · Merksatz
7:55	Projekt-Baustein	Tokenizer des Rechners erkennt −3.5 · Steuerwerk der Mini-CPU: holen, decodieren, ausführen
8:20	Prüfungsfallen	fünf Fehlvorstellungen (siehe 04)
9:14	Zusammenfassung	Kernaussagen · nicht zu schwierig, sondern unmöglich
9:51	Quiz	4 Fragen · Endkarte Folge 9

04 Typische Fehlvorstellungen

- Ein endlicher Automat kann sich beliebig viel merken – nur so viel, wie er Zustände hat; beliebig tiefe Klammern schafft er nicht.
- Ein Automat ohne Endzustand ist falsch gezeichnet – Steuerungen (Drehkreuz) brauchen keinen, nur erkennende Automaten.
- Das Halteproblem ist bloss noch ungelöst – es ist bewiesen unentscheidbar, unabhängig von Rechenleistung.
- Dann weiss man bei keinem Programm, ob es anhält – für einzelne Programme oft schon, nur gibt es kein Verfahren für alle.
- `[0-9]+` erkennt jede Zahl – nur Ziffernfolgen; Vorzeichen und Dezimaltrenner fehlen.

05 Einsatz im Unterricht

Einstieg: Die Klasse spielt das Drehkreuz nach (eine Person ist der Automat, die anderen geben Eingaben) und zeichnet danach das Diagramm.

Werkzeuge: Reguläre Ausdrücke in regex101.com testen; Automaten und Turingmaschinen in einem Online-Simulator (z. B. JFLAP oder turingmachine.io) nachbauen.

Beweis als Rollenspiel: Zwei Lernende spielen Halte-Tester und Trotzkopf – die Klasse merkt selbst, dass jede Antwort falsch wird.

Vertiefung: Aufgabe 5 (Turingmaschine von Hand) als Partnerarbeit, Aufgabe 6 als schriftliche Begründung.

Hinweise: Das Programm im Einstieg ist die Collatz-Folge; ob sie für jede Startzahl endet, ist bis heute unbewiesen. Turing bewies 1936 die Unentscheidbarkeit des Entscheidungsproblems; die Bezeichnung «Halteproblem» kam erst in den 1950er-Jahren auf. Dass eine Turingmaschine einen Laptop nachbilden kann (bei genug Speicher), ist beweisbar; die Church-Turing-These geht weiter – alles, was sich überhaupt durch ein Verfahren berechnen lässt – und ist nicht beweisbar, aber allgemein anerkannt. Die Aussage zu Virenscannern folgt aus dem Satz von Rice. Der Getränkeautomat ist ein Automat mit Ausgabe (Mealy-Automat). Reguläre Ausdrücke in Programmiersprachen können teils mehr als reguläre Sprachen (Rückverweise); im Video geht es um den regulären Kern.

06 Quellen

- A. M. Turing: On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 2. Serie, Band 42 (1936)
- J. E. Hopcroft, R. Motwani, J. D. Ullman: Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit (Pearson)
- Python-Dokumentation: re – Regular expression operations (docs.python.org/3/library/re.html)
- EDK: Rahmenlehrplan für das obligatorische Fach Informatik an Gymnasien (2017)

NAME _____ KLASSE _____ DATUM _____

Automaten und Grenzen der Berechenbarkeit

1 Drehkreuz ergänzen ●○○

Im Zustand OFFEN wirft jemand noch eine Münze ein. **Ergänze** den fehlenden Übergang im Diagramm aus dem Video und **begründe**, was sinnvollerweise passieren soll.

2 Durch 3 teilbar? ●●○

Regel: neuer Rest = (2 · alter Rest + Bit) mod 3, Start bei Rest 0. **Notiere** die Reste für a) `1111` (15) b) `1010` (10) c) `10010` (18). Welche Eingaben werden akzeptiert?

3 Muster lesen ●●○

Welche Wörter passen genau? a) `^ha+$`: ha · h · haaa b) `^Fotos?$`: Foto · Fotos · Fotoss c) `^[0-9]{2,3}$`: 7 · 42 · 123 · 1234

4 Muster schreiben ●●○

Schreibe einen regulären Ausdruck für a) eine Uhrzeit wie `07:45` (zwei Ziffern, Doppelpunkt, zwei Ziffern) b) eine Handynummer wie `079 123 45 67`. Welche falschen Eingaben lässt dein Muster in a) noch durch?

5 Turingmaschine von Hand

Die +1-Maschine aus dem Video (liest 1 → schreibt 0, links · liest 0 → schreibt 1, HALT · liest leer → schreibt 1, HALT) startet auf dem rechten Ende von `111` (7). **Notiere** Band und Kopfposition nach jedem Schritt. Was steht am Ende auf dem Band?

6 Warum es keinen Halte-Tester gibt

Erkläre in eigenen Worten, warum der Trotzkopf beweist, dass es keinen perfekten Halte-Tester geben kann. Warum hilft auch ein tausendmal schnellerer Computer nicht?

Lösungen

1 Drehkreuz ergänzen

Schleife an OFFEN mit «Münze»: Das Drehkreuz bleibt offen (die zweite Münze wird zurückgegeben oder verfällt). Ohne den Übergang wäre unklar, was in diesem Fall passiert – ein vollständiger Automat legt für jeden Zustand und jede Eingabe fest, wohin es geht.

2 Durch 3 teilbar?

a) 1, 0, 1, 0 → akzeptiert (15 = 3 · 5) · b) 1, 2, 2, 1 → abgelehnt (10 : 3 = 3 Rest 1) · c) 1, 2, 1, 0, 0 → akzeptiert (18 = 3 · 6)

3 Muster lesen

a) ha, haaa (h nicht: + verlangt mindestens ein a) · b) Foto, Fotos · c) 42, 123 (7 hat zu wenige, 1234 zu viele Ziffern)

4 Muster schreiben

a) `^[0-9]{2}:[0-9]{2}$` – lässt auch 99:99 durch; genauer: `^([01][0-9]|2[0-3]):[0-5][0-9]$` · b) `^07[0-9] [0-9]{3} [0-9]{2} [0-9]{2}$` (Schweizer Handynummern beginnen mit 07)

5 Turingmaschine von Hand

_111 (Kopf rechts) → _110 (Kopf Mitte) → _100 → _000 (Kopf auf dem leeren Feld) → 1000, HALT. Ergebnis 1000 = 8 = 7 + 1.

6 Warum es keinen Halte-Tester gibt

Der Trotzkopf fragt den Tester nach sich selbst und tut das Gegenteil: Sagt der Tester «hält an», läuft er ewig; sagt er «läuft ewig», hält er sofort an. Jede Antwort ist also falsch – ein perfekter Tester führt zu einem Widerspruch und kann nicht existieren. Das ist ein logisches Problem, kein Tempoproblem; mehr Rechenleistung ändert nichts.

Q Quiz im Video

1 A (Zustände und Übergänge) · 2 C (genau vier Ziffern) · 3 B (beliebig tiefe Klammern prüfen) · 4 C (nein, unentscheidbar)