

66115 Frühjahr 2020

Theoretische Informatik / Algorithmen (vertieft)

Aufgabenstellungen mit Lösungsvorschlägen

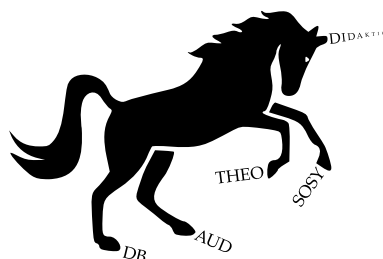


Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht

Thema Nr. 1	3
Aufgabe 2 [Automaten mit Zuständen q, r, s, t]	3
Aufgabe 3 [Kontextfreie Sprachen]	5
 Thema Nr. 2	7
Aufgabe 3 [Kontextfreie Sprachen]	7
Aufgabe 4 [Berechenbarkeitstheorie]	8
Aufgabe 8 [Nächstes rot-blaues Paar auf der x-Achse]	9
Aufgabe 10 [Niedrigster gemeinsame Vorfahre]	10



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

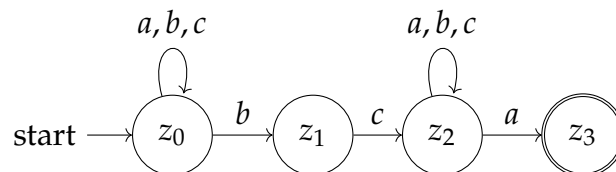


Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Thema Nr. 1

Aufgabe 2 [Automaten mit Zuständen q, r, s, t]

- (a) Es sei $L \subseteq \{a, b, c\}^*$ die von dem folgenden nichtdeterministischen Automaten akzeptierte Sprache:



Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Apmac9bwc

Beschreiben Sie (in Worten) wie die Wörter aus der Sprache L aussehen.

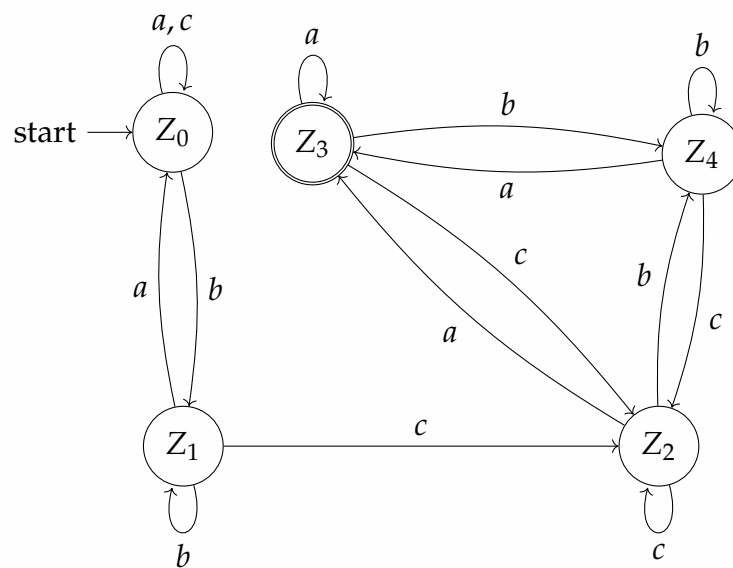
Lösungsvorschlag

Alle Wörter der Sprache L enthalten die Symbolfolge bc und enden auf a . Am Anfang der Wörter und vor dem letzten a können beliebige Kombination aus a, b, c vorkommen.

- (b) Benutzen Sie die Potenzmengenkonstruktion, um einen deterministischen Automaten zu konstruieren, der zu dem Automaten aus Teil (a) äquivalent ist. (Berechnen Sie nur erreichbare Zustände.)

Lösungsvorschlag

Zustandsmenge	Eingabe a	Eingabe b	Eingabe c
$Z_0 \{z_0\}$	$Z_0 \{z_0\}$	$Z_1 \{z_0, z_1\}$	$Z_0 \{z_0\}$
$Z_1 \{z_0, z_1\}$	$Z_0 \{z_0\}$	$Z_1 \{z_0, z_1\}$	$Z_2 \{z_0, z_2\}$
$Z_2 \{z_0, z_2\}$	$Z_3 \{z_0, z_2, z_3\}$	$Z_4 \{z_0, z_1, z_2\}$	$Z_2 \{z_0, z_2\}$
$Z_3 \{z_0, z_2, z_3\}$	$Z_3 \{z_0, z_2, z_3\}$	$Z_4 \{z_0, z_1, z_2\}$	$Z_2 \{z_0, z_2\}$
$Z_4 \{z_0, z_1, z_2\}$	$Z_3 \{z_0, z_2, z_3\}$	$Z_4 \{z_0, z_1, z_2\}$	$Z_2 \{z_0, z_2\}$



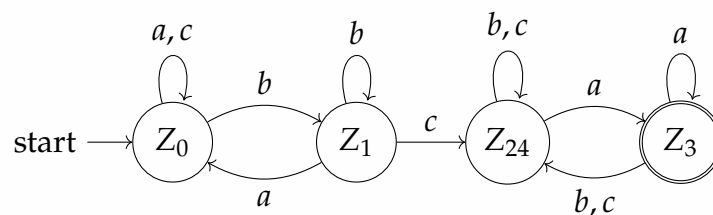
Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/A5o6pho8c

- (c) Ist der resultierende deterministische Automat schon minimal? Begründen Sie Ihre Antwort.

Lösungsvorschlag

Nein. $Z_2 \{z_0, z_2\}$ und $Z_4 \{z_0, z_1, z_2\}$ können vereinigt werden, da sie bei denselben Eingaben auf die selben Potenzenmengen übergehen.

Zustandsmenge	Eingabe a	Eingabe b	Eingabe c
$Z_2 \{z_0, z_2\}$	$Z_3 \{z_0, z_2, z_3\}$	$Z_4 \{z_0, z_1, z_2\}$	$Z_2 \{z_0, z_2\}$
$Z_3 \{z_0, z_2, z_3\}$	$Z_3 \{z_0, z_2, z_3\}$	$Z_4 \{z_0, z_1, z_2\}$	$Z_2 \{z_0, z_2\}$
$Z_4 \{z_0, z_1, z_2\}$	$Z_3 \{z_0, z_2, z_3\}$	$Z_4 \{z_0, z_1, z_2\}$	$Z_2 \{z_0, z_2\}$



Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Ai1hox2b7

- (d) Minimieren Sie den folgenden deterministischen Automaten:

Aufgabe 3 [Kontextfreie Sprachen]

- (a) Entwerfen Sie eine kontextfreie Grammatik für die folgende kontextfreie Sprache über dem Alphabet $\Sigma = \{a, b, c\}$:

$$L = \{a^{3n+2}wvc^n \mid n \in \mathbb{N}_0, 2 \cdot |w|_b = |v|_a\}$$

(Hierbei bezeichnet $|u|_x$, die Anzahl des Zeichens x in dem Wort u .)

Erklären Sie den Zweck der einzelnen Nichtterminale (Variablen) und der Grammatikregeln Ihrer Grammatik.

Lösungsvorschlag

$$P = \left\{ \begin{array}{l} S \rightarrow aaaSc \mid aaaAc \\ A \rightarrow aaB \\ B \rightarrow bBaa \mid baa \end{array} \right\}$$

Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Ghhs1xexw

- (b) Betrachten Sie die folgende kontextfreie Grammatik

$$G = (\{A, B, C, D\}, \{a, b, c\}, P, A)$$

mit den Produktionen

$$P = \left\{ \begin{array}{l} A \rightarrow AB \mid CD \mid a \\ B \rightarrow CC \mid c \\ C \rightarrow DC \mid CB \mid b \\ D \rightarrow DB \mid a \end{array} \right\}$$

Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Gf7556jn2

Benutzen Sie den Algorithmus von Cocke-Younger-Kasami (CYK), um zu zeigen, dass das Wort $abcab$ zu der von G erzeugten Sprache $L(G)$ gehört.

a	b	c	a	b
A,D	C	B	A,D	C
C	C	-	C	
C,C	A	-		
A,A	B			
A,D,B,B				

$\Rightarrow abcab \in L(G)$

- (c) Finden Sie nun ein größtmögliches Teilwort von $abcab$, dass von keinem der vier Nicht-terminale von G ableitbar ist.
- (d) Geben Sie eine Ableitung des Wortes $abcab$ mit G an.

$A \vdash AB \vdash ACC \vdash ACBC \vdash ACBDC \vdash aCBDC \vdash abBDC \vdash abcDC \vdash abcaC \vdash abcab$
--

- (e) Beweisen Sie, dass die folgende formale Sprache über $Z = a,b$ nicht kontextfrei ist: $L = a^n b^n$.

Thema Nr. 2

Aufgabe 3 [Kontextfreie Sprachen]

- (a) Ist die folgende Sprache $L_1 = \{a^{n+2}b^{2n+1} \mid n \geq 2\}$ über dem Alphabet $\Sigma = \{a, b\}$ kontextfrei?

Falls ja, geben Sie eine kontextfreie Grammatik für L_1 , an, falls nein, eine kurze Begründung (ein vollständiger Beweis ist hier nicht gefordert).

Lösungsvorschlag

L_1 ist kontextfrei

$G = (\{S, A, B\}, \{a, b\}, P, S)$

$P = \{$

$S \rightarrow aAbb$

$A \rightarrow aAbb \mid aBbb$

$B \rightarrow aab$

$\}$

Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Grxk1oczg

$n = 2$ 4a 5b: aaaabbbbb

$n = 3$ 5a 7b: aaaaabbbbbbb

$n = 4$ 6a 9b: aaaaaabbbbbbbb

- (b) Geben Sie einen Kellerautomaten (PDA) formal an, der die Sprache

$L_1 = \{w_1w_2w_3 \mid w_1, w_2, w_3 \in \Sigma^* \setminus \{\lambda\} \text{ und } w_1 = w_3^{\text{rev}}\} \in CFL$ über dem Alphabet $\Sigma = \{0, 1\}$ akzeptiert.

Dabei bezeichnet λ das leere Wort und w_3^{rev} bezeichnet das Wort w_3 rückwärts gelesen. Bei Akzeptanz einer Eingabe soll sich der PDA in einem Endzustand befinden und der Keller geleert sein.

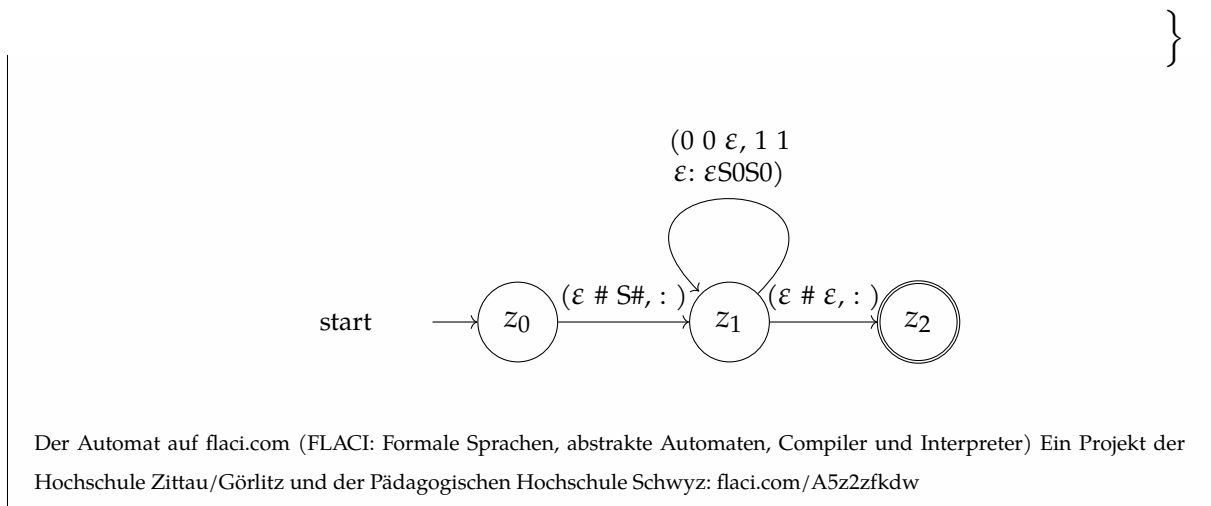
Lösungsvorschlag

Der Automat auf flaci.com (FLACI: Formale Sprachen, abstrakte Automaten, Compiler und Interpreter) Ein Projekt der Hochschule Zittau/Görlitz und der Pädagogischen Hochschule Schwyz: flaci.com/Gpkctmk3g

$P = \{$

$S \rightarrow 0S0 \mid 1S1 \mid 0A0 \mid 1A1$

$A \rightarrow 0A \mid 1A \mid 0 \mid 1$



(c) Beschreiben Sie in Worten die Arbeitsweise Ihres PDA aus Aufgabenteil (b).

Aufgabe 4 [Berechenbarkeitstheorie]

$A = \{ (M) \mid M \text{ ist Turingmaschine, die bei Eingabe } 101 \text{ hält} \}$. Dabei bezeichnet (M) die Gödelnummer der Turingmaschine M .

(a) Zeigen Sie, dass A unentscheidbar ist.

Lösungsvorschlag

Reduktionsbeweis von $H_0 \leq A$: TM U

- (i) Die zu $\square' \in \square_0$ passende TM $\square * \in \square$ aus A suchen mit $\langle \square' \rangle = \langle \square * \rangle$
- (ii) 101 auf das Band schreiben
- (iii) $\square *$ auf 101 starten

Damit könnte U H_0 entscheiden, was aber ein Widerspruch zu H_0 semi-entscheidbar ist. Damit ist A ebenfalls semi-entscheidbar.

(b) Zeigen Sie, dass A semi-entscheidbar ist.

Lösungsvorschlag

siehe a)

(c) Ist das Komplement A^c von A entscheidbar? Ist es semi-entscheidbar? Begründen Sie Ihre Antworten.

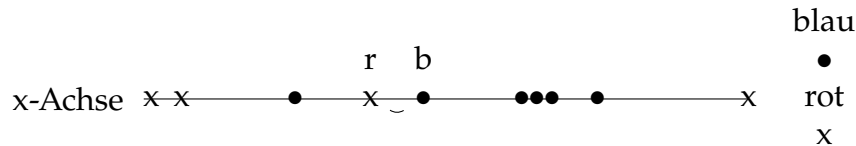
Hinweis: Sie können die Aussagen aus Teilaufgabe a) und b) verwenden, auch wenn Sie sie nicht bewiesen haben.

Lösungsvorschlag

Wenn A unentscheidbar ist, dann kann entweder A oder A^c semi-entscheidbar sein. Wären beide semi-entscheidbar, dann wäre A aber ebenfalls entscheidbar, was aber nach Voraussetzung ausgeschlossen ist.

Aufgabe 8 [Nächstes rot-blaues Paar auf der x-Achse]

Gegeben seien zwei nichtleere Mengen R und B von roten bzw. blauen Punkten auf der x-Achse. Gesucht ist der minimale euklidische Abstand $d(r, b)$ über alle Punktepaare (r, b) mit $r \in R$ und $b \in B$. Hier ist eine Beispielinstanz:



Die Eingabe wird in einem Feld A übergeben. Jeder Punkt $A[i]$ mit $1 \leq i \leq n$ hat eine x-Koordinate $A[i].x$ und eine Farbe $A[i].color \in \{\text{rot}, \text{blau}\}$. Das Feld A ist nach x-Koordinate sortiert, es gilt $A[1].x < A[2].x < \dots < A[n].x$, wobei $n = |R| + |B|$.

- (a) Geben Sie in Worten einen Algorithmus an, der den gesuchten Abstand in $\mathcal{O}(n)$ Zeit berechnet.

Lösungsvorschlag

Pseudo-Code

Algorithmus 1: Minimaler Euklidischer Abstand

```
 $d_{\min} := \max;$  // Setze  $d_{\min}$  zuerst auf einen maximalen Wert.  
for  $i$  in  $0 \dots \text{vorletzter Index}$  do ; // Iteriere über die Indizes des Punkte-Arrays  $P$  bis  
zum vorletzten Index  $P[n-1]$   
  
    if  $P[n].color \neq P[n+1].color$  then ; // Berechne den Abstand nur, wenn die Punkte  
    unterschiedliche Farben haben  
  
         $d = P[n+1].x - P[n].x$   
        if  $d < d_{\min}$  then  
             $d_{\min} = d$   
        end  
    end  
end
```

Java

```
public double findMinimalDistance() {  
    double distanceMin = Double.MAX_VALUE;  
    for (int i = 0; i < latestIndex - 1; i++) {  
        if (points[i].color != points[i + 1].color) {  
            double distance = points[i + 1].x - points[i].x;  
            if (distance < distanceMin) {  
                distanceMin = distance;  
            }  
        }  
    }  
}
```

```
return distanceMin;  
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bachelorlangaul/examen/examen_66115/jahr_2020/fruehjahr/RedBluePairCollection.java](https://github.com/src/main/java/org/bachelorlangaul/examen/examen_66115/jahr_2020/fruehjahr/RedBluePairCollection.java)

- (b) Begründen Sie kurz die Laufzeit Ihres Algorithmus.

Lösungsvorschlag

Da das Array der Länge n nur einmal durchlaufen wird, ist die Laufzeit $\mathcal{O}(n)$ sichergestellt.

- (c) Begründen Sie die Korrektheit Ihres Algorithmus.

Lösungsvorschlag

In d_{min} steht am Ende der gesuchte Wert (sofern nicht $d_{min} = Integer.MAX_VALUE$ geblieben ist)

- (d) Wir betrachten nun den Spezialfall, dass alle blauen Punkte links von allen roten Punkten liegen. Beschreiben Sie in Worten, wie man in dieser Situation den gesuchten Abstand in $\mathcal{O}(n)$ Zeit berechnen kann. (Ihr Algorithmus darf also insbesondere nicht Laufzeit $\Theta(n)$ haben.)

Lösungsvorschlag

Zuerst müssen wir den letzten blauen Punkt finden. Das ist mit einer binären Suche möglich. Wir beginnen mit dem ganzen Feld als Suchbereich und betrachten den mittleren Punkt. Wenn er blau ist, wiederholen wir die Suche in der zweiten Hälfte des Suchbereichs, sonst in der ersten, bis wir einen blauen Punkt gefolgt von einem roten Punkt gefunden haben.

Der gesuchte minimale Abstand ist dann der Abstand zwischen dem gefundenen blauen und dem nachfolgenden roten Punkt. Die Binärsuche hat eine Worst-case-Laufzeit von $\mathcal{O}(\log n)$.

Aufgabe 10 [Niedrigster gemeinsame Vorfahre]

Sei B ein binärer Suchbaum. In jedem Knoten v von B wird ein Schlüssel $v.key \in \mathbb{N}$ gespeichert sowie Zeiger $v.left$, $v.right$ und $v.parent$ auf sein linkes Kind, auf sein rechtes Kind und auf seinen Elternknoten. Die Zeiger sind nil , wenn der entsprechende Nachbar nicht existiert. Für zwei Knoten u und v ist wie üblich der *Abstand* die Anzahl der Kanten auf dem kürzesten Pfad von u nach v .

Für einen Knoten w von B sei $B(w)$ der Teilbaum von B mit Wurzel w . Für zwei Knoten u und v von B ist w ein *gemeinsamer Vorfahre*, wenn u und v in $B(w)$ liegen. Wir suchen den niedrigsten gemeinsamen Vorfahren $ngV(u, v)$ von u und v , also einen gemeinsamen Vorfahren w , so dass für jeden Vorfahren w von u und v gilt, dass w in $B(w)$ liegt. Wir betrachten verschiedene Szenarien, in denen Sie jeweils den niedrigsten gemeinsamen Vorfahren von u und v berechnen sollen.

Exkurs: Lowest Common Ancestor

Als Lowest Common Ancestor (LCA) oder „letzter gemeinsamer Vorfahre“ wird in der Informatik und Graphentheorie ein Ermittlungskonzept bezeichnet, das einen gegebenen gewurzelten Baum von Datenstrukturen effizient vorverarbeitet, sodass anschließend Anfragen nach dem letzten gemeinsamen Vorfahren für beliebige Knotenpaare in konstanter Zeit beantwortet werden können.^a

^ahttps://de.wikipedia.org/wiki/Lowest_Common_Ancestor

- (a) Wir bekommen u und v als Zeiger auf die entsprechenden Knoten in B geliefert. Beschreiben Sie in Worten und in Pseudocode einen Algorithmus, der den niedrigsten gemeinsamen Vorfahren von u und v berechnet. Analysieren Sie die Laufzeit Ihres Algorithmus.

```
/**
 * NBV = Niedrigster gemeinsamer Vorfahre.
 *
 * https://afteracademy.com/blog/lowest-common-ancestor-of-a-binary-tree
 */
public class NGV {
    static class Knoten {
        int schlüssel;
        Knoten links;
        Knoten rechts;

        public Knoten(int schlüssel) {
            this.schlüssel = schlüssel;
        }
    }

    /**
     * ngV = niedrigster gemeinsamer Vorfahre
     *
     * @param wurzel Der Wurzelknoten des Binärbaums.
     * @param knoten1 Der erste Knoten, dessen niedrigster gemeinsamer Vorfahre
     *                gesucht werden soll.
     * @param knoten2 Der zweite Knoten, dessen niedrigster gemeinsamer Vorfahre
     *                gesucht werden soll.
     *
     * @return Der niedrigste gemeinsame Vorfahre der Knoten 1 und 2.
     */
    public static Knoten ngVRekursiv(Knoten wurzel, Knoten knoten1, Knoten knoten2) {
        if (wurzel == null)
            return null;
        if (wurzel.equals(knoten1) || wurzel.equals(knoten2))
            return wurzel;
        Knoten links = ngVRekursiv(wurzel.links, knoten1, knoten2);
        Knoten rechts = ngVRekursiv(wurzel.rechts, knoten1, knoten2);
        if (links == null)
            return rechts;
        else if (rechts == null)
            return links;
        else
            return wurzel;
    }
}
```

```

}

/**
 * <pre>
 * {@code
 *      20
 *     / \
 *    8   22
 *   / \
 *  4   12
 *   / \
 *  10  14
 * }
 * </pre>
 *
 * Beispiele von
 * https://www.geeksforgeeks.org/lowest-common-ancestor-in-a-binary-search-tree/
 *
 * @param args Kommandozeilen-Argumente
 */
public static void main(String[] args) {
    Knoten wurzel = new Knoten(20);

    Knoten knoten8 = new Knoten(8);
    Knoten knoten22 = new Knoten(22);
    Knoten knoten4 = new Knoten(4);
    Knoten knoten12 = new Knoten(12);
    Knoten knoten10 = new Knoten(10);
    Knoten knoten14 = new Knoten(14);

    wurzel.links = knoten8;
    wurzel.rechts = knoten22;
    wurzel.links.links = knoten4;
    wurzel.links.rechts = knoten12;
    wurzel.links.rechts.links = knoten10;
    wurzel.links.rechts.rechts = knoten14;

    System.out.println(ngVRekursiv(wurzel, knoten10, knoten14).schlüssel); // 12
    System.out.println(ngVRekursiv(wurzel, knoten14, knoten8).schlüssel); // 8
    System.out.println(ngVRekursiv(wurzel, knoten10, knoten22).schlüssel); // 20
}
}

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_66115/jahr_2020/fruehjahr/NGV.java](https://github.com/orgs/bschlangaul/examen/examen_66115/jahr_2020/fruehjahr/NGV.java)

- (b) Wir bekommen u und v wieder als Zeiger auf die entsprechenden Knoten in B geliefert. Seien d_u und d_v die Abstände von u bzw. v zum niedrigsten gemeinsamen Vorfahren von u und v . Die Laufzeit Ihres Algorithmus soll $O(\max\{d_u, d_v\})$ sein. Dabei kann Ihr Algorithmus in jedem Knoten v eine Information $v.info$ speichern. Skizzieren Sie Ihren Algorithmus in Worten.
- (c) Wir bekommen die Schlüssel $u.key$ und $v.key$. Die Laufzeit Ihres Algorithmus soll proportional zum Abstand der Wurzel von B zum niedrigsten gemeinsamen Vorfahren von u und v sein. Skizzieren Sie Ihren Algorithmus in Worten.