Arbeitsheft 2: NP-Vollständigkeit

(BuK / WS 2018 / RWTH Aachen)

Gerhard J. Woeginger

Dieses Arbeitsheft enthält einige Übungsaufgaben zur NP-Vollständigkeit. Jede Aufgabe besteht im Wesentlichen aus einem einzigen, langen NP-Vollständigkeitsbeweis, der in viele kleine Stücke zerbrochen wurde. Wenn man sich Schritt für Schritt durch diese kleinen Stücke durcharbeitet, so entdeckt man den Beweis.

Die Aufgaben sind für jeden Informatik-Studenten im zweiten Studienjahr lösbar, der die Vorlesung über Berechenbarkeit und Komplexität (BuK) besucht hat. **Für die Aufgaben in diesem Heft werden keine Musterlösungen bereitgestellt. Die Lösungen werden weder im Tutorium noch in der Globalübung diskutiert.** Zehn Minuten eigenständiges Denken sind nützlicher, als wenn man sich fünf Stunden lang Lösungen durchliest, die von anderen erstellt wurden.

---

1 Kern-Mengen in gerichteten Graphen

Eine *Kern-Menge* in einem gerichteten Graphen $G = (V, E)$ ist eine Teilmenge $K \subseteq V$ der Knoten mit den folgenden beiden Eigenschaften:

- Keine zwei Knoten in K sind mit einander durch einen Pfeil in E verbunden.
- Von jedem Knoten $v \notin K$ geht ein Pfeil $v \to k$ in E aus, der zu einem Knoten $k \in K$ führt.

Einige gerichtete Graphen besitzen eine Kern-Menge, für andere gerichtete Graphen gibt es keine Kern-Menge. Die folgenden drei Teilaufgaben sollen Sie mit Kern-Mengen vertraut machen.

(a) Zeigen Sie, dass der Graph mit $V = \{1, 2, \ldots, n\}$ und $E = \{i \to j \mid i \neq j\}$ eine Kern-Menge besitzt.

(b) Der gerichtete Kreis C_n hat $V = \{1, 2, \ldots, n\}$, und Pfeile $i \to i + 1$ für $1 \leq i \leq n - 1$, und einen Pfeil $n \to 1$. Bestimmen Sie alle $n \geq 3$, für die C_n eine Kern-Menge besitzt.

(c) Ein *orientierter Baum* entsteht aus einem (ungericheten) Baum, indem man jede Kante $\{u, v\}$ durch einen der beiden Pfeile $u \to v$ oder $v \to u$ ersetzt. Beweisen oder widerlegen Sie: Jeder orientierte Baum besitzt eine Kern-Menge.

Nun wenden wir uns der Komplexität zu. Das Entscheidungsproblem KERN-MENGE nimmt als Eingabe einen gerichteten Graphen $G = (V, E)$, und fragt, ob G eine Kern-Menge besitzt. Wir werden zeigen, dass das Problem KERN-MENGE NP-vollständig ist. Dazu müssen wir (erstens) zeigen, dass KERN-MENGE in NP liegt und (zweitens), dass KERN-MENGE NP-hart ist. Der erste Schritt ist nicht so schwer:

(d) Zeigen, Sie, dass das Problem KERN-MENGE in der Klasse NP liegt. Wie sieht ein kurzes und einfaches Zertifikat für Ja-Instanzen aus? Wie verifiziert man dieses Zertifikat in polynomieller Zeit?

Für den NP-Schwere-Beweis zeigen wir, dass 3-SAT auf KERN-MENGE polynomiell reduzierbar ist: 3-SAT $\leq$ KERN-MENGE. Die Reduktion beginnt mit einer beliebigen Instanz von 3-SAT, also mit einer beliebigen Boole'schen Formel Φ in 3-CNF über einer Boole'schen Variablenmenge $X = \{x_1, \ldots, x_n\}$. Diese Formel Φ übersetzen wir in einen gerichteten Graphen.

(e) Führen Sie für jede Variable x_i zwei entsprechende Knoten $A(x_i)$ und $A(\overline{x_i})$ ein, die dem positiven und dem negativen Literal entsprechen. Fügen Sie die beiden Pfeile $A(x_i) \to A(\overline{x_i})$ und $A(\overline{x_i}) \to A(x_i)$ hinzu. Ist es möglich, dass eine Kern-Menge gleichzeitig den Knoten $A(x_i)$ und den Knoten $A(\overline{x_i})$ enthält?

(f) Im Laufe der Reduktion werden wir weitere Knoten und weitere Pfeile einführen. Einige dieser Pfeile werden auf $A(x_i)$ und/oder auf $A(\overline{x_i})$ zeigen, aber keiner dieser Pfeile wird von $A(x_i)$ oder $A(\overline{x_i})$ ausgehen. Ist es unter diesen Umständen möglich, dass eine Kern-Menge weder den Knoten $A(x_i)$ noch den Knoten $A(\overline{x_i})$ enthält?

(g) Zurück zur Reduktion. Führen Sie für jede Klausel c in Φ drei entsprechende Knoten $B_1(c)$, $B_2(c)$, $B_3(c)$ ein, die einen gerichteten Kreis $B_1(c) \to B_2(c) \to B_3(c) \to B_1(c)$ bilden.

Wenn die Klausel c das Literal x_i enthält, dann kreieren Sie die drei Pfeile $B_1(c) \to A(x_i)$ und $B_2(c) \to A(x_i)$ und $B_3(c) \to A(x_i)$. Wenn die Klausel c das Literal $\overline{x_j}$ enthält, dann kreieren Sie die drei Pfeile $B_1(c) \to A(\overline{x_j})$ und $B_2(c) \to A(\overline{x_j})$ und $B_3(c) \to A(\overline{x_j})$.

Damit ist unsere Reduktion von 3-SAT auf KERN-MENGE abgeschlossen. Den resultierenden gerichteten Graphen nennen wir $G(\Phi)$. Wir müssen zeigen, dass die Reduktion in polynomieller Zeit durchgeführt werden kann und dass die Reduktion wirklich korrekt ist. Wir beginnen mit der Zeitanalyse.

(h) Angenommen, die Boole'sche Formel Φ besteht aus m Klauseln mit insgesamt n Variablen. Wieviele Knoten hat dann der von Ihnen konstruierte Graph $G(\Phi)$? Ist die Anzahl der Knoten polynomiell in der Beschreibungslänge $m + n$ von Φ beschränkt? Warum? Warum nicht?

(i) Wieviele Pfeile gibt es im Graphen $G(\Phi)$? Ist die Anzahl der Pfeile polynomiell in $m+n$ beschränkt?

(j) Hier ist eine nützliche allgemeine Beobachtung (die nicht nur im momentanen Beweis verwendet werden kann): Wenn in einem Graphen die Anzahl der Knoten polynomial in x beschränkt ist (wobei x eine Eingabelänge misst), dann ist automatisch auch die Anzahl der Kanten/Pfeile polynomial in x beschränkt. Begründen Sie diese Beobachtung.

Da jeder einzelne Schritt in der Reduktion einen neuen Knoten oder einen neuen Pfeil in $G(\Phi)$ erzeugt, können wir jetzt folgern, dass auch die Gesamtzeit der Reduktion polynomiell in $m+n$ beschränkt ist. Anmerkung: Wir haben die Zeitanalyse sehr detailliert und äusserst ausführlich durchgeführt. In den meisten Fällen (und auch bei der Klausur) genügt eine kurze (und präzise) Diskussion.

Nun zur Korrektheit. Wir wollen zeigen, dass die Formel Φ dann und nur dann eine erfüllende Wahrheitsbelegung hat, wenn der von Ihnen konstruierte Graph $G(\Phi)$ eine Kern-Menge besitzt. Wir beginnen mit der einfachen Richtung der dann-und-nur-dann Aussage. Wir nehmen an, dass es eine Wahrheitsbelegung W von $x_1, \ldots, x_n$ gibt, die die Formel Φ erfüllt.

(k) Wenn ein Literal x_i oder $\overline{x_i}$ von der Wahrheitsbelegung W wahr gemacht wird, so geben wir den entsprechenden Knoten $A(x_i)$ oder $A(\overline{x_i})$ in eine Menge K. Zeigen Sie: Diese Menge K bildet eine Kern-Menge für den Graphen $G(\Phi)$.

Nun kommen wir zum aufwendigeren Teil der dann-und-nur-dann Aussage. Wir nehmen an, dass der Graph $G(\Phi)$ eine Kern-Menge K besitzt.

(l) Zeigen Sie: Für jede Boole'sche Variable x_i liegt genau einer der beiden Knoten $A(x_i)$ und $A(\overline{x_i})$ in K.

(m) Zeigen Sie: Für jede Klausel c liegt höchstens einer der drei Knoten $B_1(c)$, $B_2(c)$, $B_3(c)$ in K.

(n) Zeigen Sie: Besteht die Klausel c aus den drei Literalen ℓ_1, ℓ_2, ℓ_3, so liegt mindestens einer der drei Knoten $A(\ell_1), A(\ell_2), A(\ell_3)$ in K.

(o) Mit Hilfe der obigen Beobachtungen kann man nun aus K eine erfüllende Wahrheitsbelegung für Φ ablesen. Wie sieht diese Wahrheitsbelegung aus?

Damit sind beide Richtungen des Korrektheitsbeweises gezeigt.

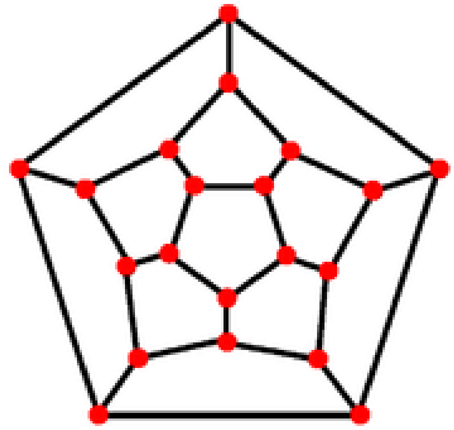

Figure 1: Wie sieht die 3-Färbung aus?

2 Drei-Färbbarkeit von Graphen

Ein Graph $G = (V, E)$ ist 3-*färbbar*, wenn seine Knoten mit drei Farben eingefärbt werden können, sodass benachbarte Knoten immer verschiedene Farben erhalten. Die drei Farben nennen wir der Einfachheit halber 1,2,3.

(a) Der ungerichtete Kreis C_n hat $V = \{1, 2, \ldots, n\}$, und Kanten $\{i, i+1\}$ für $1 \leq i \leq n-1$, und eine Kante $\{1, n\}$. Zeigen Sie: Für jedes $n \geq 3$ ist der Kreis C_n 3-färbbar.

(b) Zu einem Baum $T = (V, E)$ fügen wir eine Kante zwischen zwei Knoten u und v mit $u \neq v$ hinzu. Zeigen Sie: Der resultierende Graph ist 3-färbbar.

(c) Zeigen Sie: Der Graph in Abbildung 1 ist 3-färbbar.

Nun wenden wir uns der Komplexität des entsprechenden Entscheidungsproblems zu. Das 3-COLORING Problem nimmt als Eingabe einen ungerichteten Graphen $G = (V, E)$, und fragt, ob es eine Färbung $f : V \to \{1, 2, 3\}$ gibt, sodass alle Kanten $\{u, v\} \in E$ die Bedingung $f(u) \neq f(v)$ erfüllen.

Wir werden zeigen, dass das 3-COLORING Problem NP-vollständig ist. Der erste Schritt besteht wieder darin, zu zeigen, dass 3-COLORING in NP liegt.

(d) Zeigen, Sie, dass das Problem 3-COLORING in der Klasse NP liegt. Wie sieht ein kurzes und einfaches Zertifikat für Ja-Instanzen aus? Wie verifiziert man dieses Zertifikat in polynomieller Zeit?

Der zweite Schritt ist wieder der NP-Schwere-Beweis. In diesem Schwere-Beweis werden wir das wie folgt definierte *Gadget* G_9 mit der Knotenmenge $\{\alpha, \beta, \gamma, y_1, y_2, y_3, y_4, y_5, y_6\}$ verwenden: G_9 enthält zwei Dreiecke y_1, y_2, y_3 und y_4, y_5, y_6, die mit einander durch die Kante $[y_1, y_4]$ verbunden sind. Weiters enthält G_9 die drei Kanten $[\alpha, y_2]$, $[\beta, y_3]$ und $[\gamma, y_5]$.

(e) Skizzieren Sie den Gadget-Graphen G_9.

(f) Zeigen Sie: Wenn in einer 3-Färbung von G_9 die drei Knoten α, β, γ dieselbe Farbe f erhalten, dann muss auch der Knoten y_6 diese Farbe f erhalten.

(g) Zeigen Sie: Wenn eine Färbung der drei Knoten α, β, γ mindestens einem Knoten die Farbe f zuweist, so kann diese Färbung zu einer 3-Färbung von $\{y_1, y_2, y_3, y_4, y_5, y_6\}$ erweitert werden, die dem Knoten y_6 die Farbe f zuweist.

Nun werden wir zeigen, dass 3-SAT polynomiell auf 3-COLORING reduzierbar ist: 3-SAT $\leq$ 3-COLORING. Die Reduktion beginnt wieder mit einer beliebigen Instanz von 3-SAT, also mit einer beliebigen Boole'schen Formel Φ in 3-CNF über einer Boole'schen Variablenmenge $X = \{x_1, \ldots, x_n\}$. Diese Formel Φ übersetzen wir nun in einen Graphen $G(\Phi)$.

(h) Der Graph $G(\Phi)$ enthält zunächst einmal ein Dreieck mit den drei Knoten WAHR, FALSCH und DUMMY. Zeigen Sie: Falls $G(\Phi)$ eine 3-Färbung besitzt, so müssen die drei Knoten WAHR, FALSCH und DUMMY drei verschiedene Farben erhalten. Von jetzt an bezeichnen wir die Farbe des Knotens WAHR mit WAHR und die Farbe des Knotens FALSCH mit FALSCH.

(i) Führen Sie für jede Variable x_i in der 3-SAT Instanz zwei entsprechende Knoten $A(x_i)$ und $A(\overline{x_i})$ ein, die dem positiven und dem negativen Literal entsprechen. Fügen Sie die Kante zwischen $A(x_i)$ und $A(\overline{x_i})$ hinzu, sowie die beiden Kanten zwischen DUMMY und $A(x_i)$, und zwischen DUMMY und $A(\overline{x_i})$.

Zeigen Sie, dass in jeder 3-Färbung von $G(\Phi)$ folgendes gilt: Entweder (1) $A(x_i)$ ist mit WAHR und $A(\overline{x_i})$ ist mit FALSCH gefärbt, oder (2) $A(x_i)$ ist mit FALSCH und $A(\overline{x_i})$ ist mit WAHR gefärbt.

(j) Führen Sie für jede Klausel c in Φ eine Kopie $G_9(c)$ des Gadgets G_9 ein. Wenn die Klausel c die Literale ℓ_1, ℓ_2, ℓ_3 enthält, dann verschmelzen Sie den Knoten α in $G_9(c)$ mit dem Knoten $A(\ell_1)$, den Knoten β in $G_9(c)$ mit dem Knoten $A(\ell_2)$, und den Knoten γ in $G_9(c)$ mit dem Knoten $A(\ell_3)$. Führen Sie weiters die beiden Kanten zwischen FALSCH und y_6, und zwischen DUMMY und y_6 ein.

Damit ist unsere Reduktion auch schon vollständig beschrieben. Wir wenden uns der Zeitanalyse zu.

(k) Angenommen, die Boole'sche Formel Φ besteht aus m Klauseln mit insgesamt n Variablen. Wieviele Knoten hat der von Ihnen konstruierte Graph $G(\Phi)$? Zeigen Sie, dass die Anzahl der Knoten polynomiell in der Beschreibungslänge $m + n$ beschränkt ist.

(l) Wieviele Kanten gibt es im Graphen $G(\Phi)$? Zeigen Sie, dass die Anzahl der Kanten polynomiell in der Beschreibungslänge $m + n$ beschränkt ist.

Und schlussendlich kommen wir zum Korrektheitsargument. Wir müssen zeigen, dass die Formel Φ dann und nur dann eine erfüllende Wahrheitsbelegung hat, wenn der von Ihnen konstruierte Graph $G(\Phi)$ eine 3-Färbung erlaubt. Wir nehmen zunächst an, dass es eine Wahrheitsbelegung W von $x_1, \ldots, x_n$ gibt, die die Formel Φ erfüllt.

(m) Wenn ein Literal ℓ_i von der Wahrheitsbelegung W wahr gemacht wird, so färben wir den entsprechenden Knoten $A(x_i)$ mit der Farbe WAHR. Wenn ein Literal ℓ_i von der Wahrheitsbelegung W falsch gemacht wird, so färben wir den entsprechenden Knoten $A(x_i)$ mit der Farbe FALSCH. Zeigen Sie, dass diese Teilfärbung zu einer gültigen 3-Färbung der gesamten Knotenmenge von $G(\Phi)$ erweitert werden kann.

Als nächstes nehmen wir an, dass der Graph $G(\Phi)$ eine 3-Färbung besitzt.

(n) Zeigen Sie: Für jede Klausel c ist der Knoten y_6 in $G_9(c)$ mit der Farbe WAHR gefärbt.

(o) Zeigen Sie: Wenn die Klausel c die Literale ℓ_1, ℓ_2, ℓ_3 enthält, so ist mindestens einer der drei Knoten $A(\ell_1)$, $A(\ell_2)$, $A(\ell_3)$ mit der Farbe WAHR gefärbt.

(p) Folgern Sie daraus: Wenn $G(\Phi)$ eine 3-Färbung besitzt, dann ist die Formel Φ erfüllbar.

Damit sind beide Richtungen des Korrektheitsbeweises gezeigt.

(q) Zeigen Sie: Das Problem 2-COLORING ist polynomiell lösbar.

---

3 Drei-Färbbarkeit von planaren Graphen

Ein Graph $G = (V, E)$ heisst *planar*, wenn er wie folgt in die Euklidische Ebene eingebettet werden kann: Jeder Knoten $v \in V$ wird in einen Punkt $P(v)$ eingebettet, wobei verschiedene Knoten in verschiedenen Punkten eingebettet werden. Jede Kante $\{u, v\} \in E$ wird in einen polygonalen Streckenzug $Z(u, v)$ eingebettet, der im Punkt $P(u)$ beginnt und im Punkt $P(v)$ endet. Wenn zwei Kanten in E keinen gemeinsamen Knoten enthalten, so sind ihre entsprechenden Streckenzüge disjunkt. Wenn zwei Kanten in E einen gemeinsamen Knoten v enthalten, so haben ihre Streckenzüge nur den Punkt $P(v)$ gemein. Ein zentraler Satz in der Graphentheorie besagt, dass in einem planaren Graphen $G = (V, E)$ mit $|V| \geq 2$ immer die Ungleichung $|E| \leq 3|V| - 6$ gilt.

(a) Der vollständige Graph K_n hat die Knotenmenge $V = \{1, 2, \ldots, n\}$ und enthält alle Kanten $\{i, j\}$ mit $i \neq j$. Bestimmen Sie alle $n \geq 3$, für die der Graph K_n planar ist.

(b) Zeigen Sie: Jeder Baum ist planar.

(c) Beweisen oder widerlegen Sie: Wenn ein Graph $G = (V, E)$ die Ungleichung $|E| \leq 3|V| - 6$ erfüllt, dann ist G planar.

Wir betrachten nun das PLANAR-3-COLORING Problem: Als Eingabe erhalten wir einen planaren Graphen $G = (V, E)$ mit einer Einbettung in der Euklidischen Ebene, und wir wollen wissen, ob dieser planare Graph 3-färbbar ist.

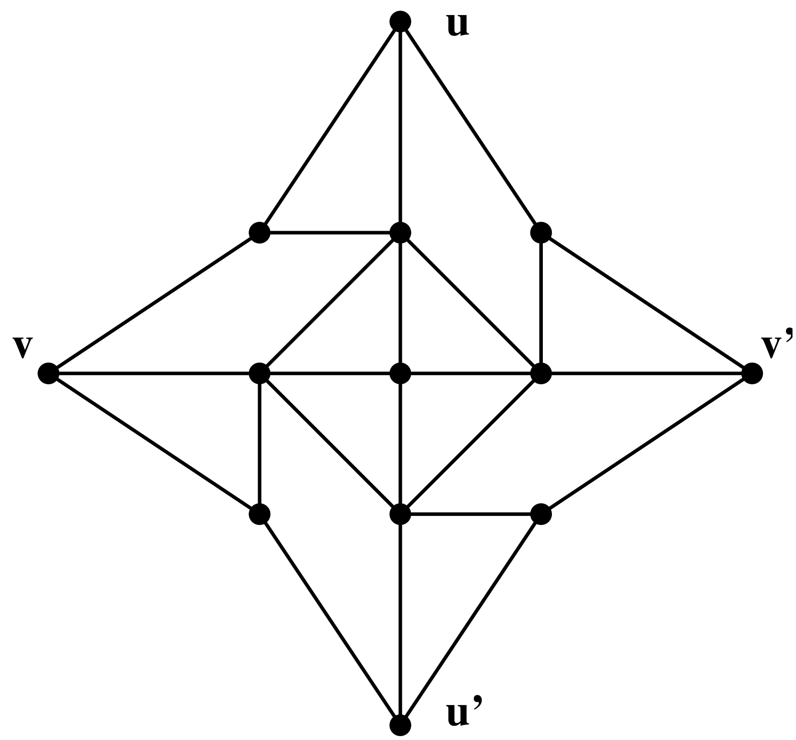

Figure 2: Das Kreuzungs-Gadget $G(u, v, u', v')$

(d) Nach unserer Vorarbeit über 3-COLORING im vorhergehenden Abschnitt ist es leicht zu sehen, dass PLANAR-3-COLORING in NP liegt. Wie sieht man das?

Unser wichtigstes Werkzeug im NP-Schwere-Beweis ist das Kreuzungs-Gadget $G(u, v, u', v')$ in Abbildung 2. Das Kreuzungs-Gadget besteht aus 13 Knoten, von denen die vier Knoten u, v, u', v' in exakt dieser Reihenfolge gegen den Uhrzeigersinn am Rand liegen.

(e) Zeigen Sie: Es gibt eine 3-Färbung von $G(u, v, u', v')$, in der alle vier Knoten u, v, u', v' die selbe Farbe erhalten.

(f) Zeigen Sie: Es gibt eine 3-Färbung von $G(u, v, u', v')$, in der u und u' die selbe Farbe f_1 erhalten und in der v und v' die selbe Farbe $f_2 \neq f_1$ erhalten.

(g) Zeigen Sie: In jeder 3-Färbung von $G(u, v, u', v')$ erhalten u und u' die selbe Farbe. In jeder 3-Färbung von $G(u, v, u', v')$ erhalten v und v' die selbe Farbe.

Im NP-Schwere-Beweis werden wir zeigen, dass 3-COLORING polynomiell auf PLANAR-3-COLORING reduzierbar ist: 3-COLORING $\leq$ PLANAR-3-COLORING. Die Reduktion beginnt mit einer beliebigen Instanz von 3-COLORING, also mit einem ungerichteten Graphen $G' = (V', E')$. Diese Instanz übersetzen wir nun in eine PLANAR-3-COLORING Instanz.

(h) Betten Sie den Graphen $G' = (V', E')$ in die Euklidische Ebene ein. (Achtung: Dieser Graph G' ist nicht notwendigerweise planar.) Die Punkte $P(v)$ für Knoten $v \in V'$

können beliebig gewählt werden. Jede Kante $\{u, v\} \in E'$ wird in einen polygonalen Streckenzug eingebettet, der im Punkt $P(u)$ beginnt und im Punkt $P(v)$ endet. Dabei dürfen die Streckenzüge von verschiedenen Kanten einander kreuzen. Durch jeden Punkt (mit Ausnahme der eingebetteten Knoten) gehen immer nur höchstens zwei dieser Streckenzüge. Argumentieren Sie, dass dies tatsächlich immer möglich ist.

(i) Im nächsten Schritt ersetzen Sie jede Kreuzung von zwei polygonalen Streckenzügen durch eine Kopie des Kreuzungs-Gadgets $G(u, v, u', v')$. Dadurch tauchen auf einmal viele neue Knoten u, v, u', v' von den verschiedenen Kopien auf. Einige dieser Knoten sollten nun mit einander verschmolzen werden. Welche Knoten werden mit einander verschmolzen? Warum?

Damit ist unsere Konstruktion abgeschlossen.

(j) Zeigen Sie: Der resultierende Graph ist planar.

(k) Zeigen Sie: Der resultierende Graph und eine planare Einbettung können in polynomieller Zeit berechnet werden.

(l) Zeigen Sie: Die 3-COLORING Instanz $G' = (V', E')$ hat eine 3-Färbung, genau dann wenn die konstruierte PLANAR-3-COLORING Instanz eine 3-Färbung hat.

Damit ist unser NP-Vollständigkeitsbeweis abgeschlossen.

---

4 Prozessplanung mit Deadlines

Wir betrachten n Prozesse $P_1, \ldots, P_n$ mit

- Prozesszeiten $t_1, \ldots, t_n$
- Deadlines $d_1, \ldots, d_n$
- Strafkosten $s_1, \ldots, s_n$.

Diese Prozesse sollen in geeigneter Reihenfolge (sequentiell) auf einem einzelnen Prozessor abgearbeitet werden. Jeder Prozess P_i ($i = 1, \ldots, n$) ist dabei ab dem Zeitpunkt $t = 0$ verfügbar und muss für t_i unmittelbar aufeinanderfolgende Zeiteinheiten bearbeitet werden. Ein Prozess P_i heisst *pünktlich*, falls er zum Zeitpunkt d_i bereits vollständig abgearbeitet ist; andernfalls nennen wir den Prozess *verspätet*. Für jeden verspäteten Prozess müssen zur Strafe s_i Euro bezahlt werden. In der Optimierungsvariante des Problems suchen wir nach einem Bearbeitungsplan, der die Gesamtsumme der bezahlten Strafkosten minimiert.

(a) Bestimmen Sie eine Optimallösung für die Instanz mit $n = 6$ Prozessen mit Prozesszeiten $1, 2, 3, 4, 5, 6$, Deadlines $1, 3, 7, 8, 13, 19$ und Strafkosten $1, 2, 3, 1, 2, 3$.

(b) Zeigen Sie: Es gibt immer eine Optimallösung, in der zuerst alle pünktlichen Prozesse und danach erst alle verspäteten Prozesse bearbeitet werden.

(c) Angenommen, in einem Bearbeitungsplan werden direkt hintereinander zwei pünktliche Prozesse J_a und J_b bearbeitet, für deren Deadlines $d_a > d_b$ gilt. Zeigen Sie, dass das Vertauschen von J_a und J_b die Gesamtsumme der bezahlten Strafkosten nicht erhöhen kann.

(d) Zeigen Sie: Es gibt immer eine Optimallösung, in der die pünktlichen Prozesse in Reihenfolge von nicht-fallenden Deadlines angeordnet sind.

(e) Formulieren Sie die Optimierungsvariante in ein geeignetes Entscheidungsproblem um. Führen Sie dazu eine Schranke S für die Gesamtsumme der bezahlten Strafkosten ein.

Das Entscheidungsproblem werden wir von nun an mit DEADLINE bezeichnen. Wir werden zeigen, dass das DEADLINE Problem NP-vollständig ist. Der erste Schritt besteht wieder einmal darin, zu zeigen, dass DEADLINE in NP liegt.

(f) Zeigen, Sie, dass das Problem DEADLINE in der Klasse NP liegt. Wie sieht ein kurzes und einfaches Zertifikat für Ja-Instanzen aus? Wie verifiziert man dieses Zertifikat in polynomieller Zeit?

Der zweite Schritt ist der NP-Schwere-Beweis. Wir werden zeigen, dass PARTITION polynomiell auf DEADLINE reduzierbar ist: PARTITION $\leq$ DEADLINE. Die Reduktion beginnt mit einer beliebigen Instanz von PARTITION, also mit m positiven ganzen Zahlen $a_1, \ldots, a_m$ mit Summe $\sum_{j=1}^{m} a_j = 2A$. Diese Instanz übersetzen wir nun in eine DEADLINE Instanz.

(g) Führen Sie für jede Zahl a_j in der Instanz von PARTITION einen entsprechenden Prozess P_j ein, mit Prozesszeit $t_j = a_j$ und Deadline $d_j = A$ und Strafkosten $s_j = a_j$.

Nun müssen wir noch die Schranke S für die Gesamtsumme der bezahlten Strafkosten geeignet wählen, sodass die Instanz von PARTITION genau dann lösbar ist, wenn die von uns konstruierte Instanz von DEADLINE lösbar ist.

(h) Bestimmen Sie einen geeigneten Wert für diese Schranke S.

(i) Zeigen Sie: Wenn die PARTITION Instanz lösbar ist, dann gibt es für die DEADLINE Instanz einen Bearbeitungsplan, in dem die Gesamtsumme der bezahlten Strafkosten höchstens S ist.

(j) Zeigen Sie: Wenn die DEADLINE Instanz einen Bearbeitungsplan besitzt, in dem die Gesamtsumme der bezahlten Strafkosten höchstens S beträgt, dann ist die PARTITION Instanz lösbar.

Damit sind beide Richtungen des Korrektheitsbeweises gezeigt und unser NP-Vollständigkeitsbeweis ist abgeschlossen.

(k) Folgt aus unserem NP-Schwere-Beweis, dass DEADLINE stark NP-schwer ist?

(l) Folgt aus unserem NP-Schwere-Beweis, dass DEADLINE <u>nicht</u> stark NP-schwer ist?