

Lösungsstrategien für NP-schwere Probleme der Kombinatorischen Optimierung

— Übungsblatt 2 —

Walter Stieben
(4stieben@inf)

Tim Reipschläger
(4reipsch@inf)

Louis Kobras
(4kobras@inf)

Hauke Stieler
(4stieler@inf)

Abgabe am: 25. April 2016

Aufgabe 2.1

Aufgabe 2.1.1

Aufgabe 2.1.2 Spezialfall $k = 2$

Beim Spezialfall für $k = 2$ kann man tatsächlich einen Algorithmus angeben, dessen Laufzeit polynomiell in der Eingabe ist.

Algorithm 1 reserve

```

1: procedure RESERVE( $P, R, k$ )
2:   for all  $p \in P$  do
3:      $R_{rest} = \text{weiseRessourcenZu}(p, R)$  // gibt alle noch nicht zugewiesenen Ressourcen zurück
4:      $P = P - p$ 
5:     for all  $q \in P$  do
6:        $b = \text{istVerteilungMöglich}(q, R_{rest})$ 
7:       if  $b == \text{true}$  then return true
8:     end if
9:   end for
10:  end for return false
11: end procedure

```

Die Laufzeit beträgt dabei $\mathcal{O}(|P| \cdot n + |P|^2 \cdot n)$, wobei $n = |R|$ ist.

Man geht in der äußeren Schleife alle $p \in P$ durch, also $|P|$ mal. Dort verteilt man zunächst maximal n viele Ressourcen, ergo n viele Schritte. Nun wird p aus P entfernt, was mit einer schlaun Implementierung (z.B. als Liste) in konstanter Zeit machbar ist.

Danach beginnt die innere Schleife, welche alle $q \in P$ durch geht und somit $|P| - 1$ mal durchläuft. Dort wird dann geprüft ob eine weitere Verteilung der restlichen Ressourcen auf q möglich ist. Auch da benötigt man maximal n viele Schritte.

Man erhält also eine Laufzeit von $\mathcal{O}(|P| \cdot (n + |P| \cdot n))$, was mit $\mathcal{O}(|P| \cdot n + |P|^2 \cdot n)$ äquivalent ist (hier erkennt man aber schön das Polynom).

Aufgabe 2.1.3

Aufgabe 2.1.4

Aufgabe 2.2

Aufgabe 2.2.1 3 – SAT auf SET SPLITTING reduzieren

Aufgabe 2.2.1.a Zeigen, dass SET SPLITTING $\in NP$ gilt

Mit einem Verifikationsalgorithmus, welcher die Eingabe $\langle S_1, S_2, C \rangle$ bekommt, kann man in polynomialer Laufzeit prüfen, ob eine Aufteilung in Klassen (also ein set splitting), korrekt ist.

Dazu geht man zunächst jedes $c \in C$ durch und nimmt ein Element e_1 aus c . Für dieses Element prüft man nun ob es ein $e_2 \in c$ gibt, bei dem gilt $class(e_1) \neq class(e_2)$ (also ob e_1 in einer anderen Klasse ist, als e_2). Gibt es ein solches e_2 , ist die Aufteilung der Teilmenge c schon mal gültig und verifiziert. Zu prüfen sind noch die restlichen Teilmengen.

Die Laufzeit ist polynomial, da man $|C|$ viele Teilmengen durch geht und pro Teilmenge maximal $|S|$ Elemente. Es gibt maximal $|S|$ viele Teilmengen (jede Teilmenge mit je einem Element), damit wäre die Laufzeit in $\mathcal{O}(|S|^2)$.

Aufgabe 2.2.1.b Reduktion angeben

Gegeben sei eine Aussagenlogische Formel A mit den Klauseln K_1, K_2, \dots, K_k und den Literalen x_1, x_2, \dots, x_n , wobei jede Klausel maximal drei Literale enthält. Um eine Eingabe für das SET SPLITTING Problem zu bekommen, wandelt man A wie folgt um:

Jedes Literal wird mit seinem Komplement in eine Menge geschlossen, also $\{x_1, \bar{x}_i\}$, analog wird jede Klausel in eine Menge geschlossen. Dazu wird ein neues Element F erzeugt und jeder bisherigen Teilmenge hinzugefügt, sodass F in jeder Teilmenge vorhanden ist. F wird als **false** interpretiert und ist immer in der Partition enthalten, in der alle Literale, welche zu 0 ausgewertet werden enthalten sind.

Damit ist die Reduktion fertig und man muss nur noch das Orakel von SET SPLITTING befragen.

Aufgabe 2.2.1.c Korrektheit für " \Rightarrow "

Behauptung: A besitzt eine erfüllende Belegung, dann existiert ein SET SPLITTING in obiger Konstruktion.

Seien $x_1^*, x_2^*, \dots, x_n^*$ die Belegungen für die Literale aus A , welche A erfüllen. Man teilt nun die Elemente der zugehörigen Mengen aus der Konstruktion so auf, dass in S_1 alle diejenigen Literale enthalten sind, die zu 1 und in S_2 diejenigen, die zu 0 ausgewertet werden. Durch das neue Element F gelingt es nun eine Partitionierung zu finden, da man zwei Fälle unterscheiden kann:

Entweder besitzt eine betrachtete Klausel mindestens ein zu einer 1 ausgewertetes und mindestens ein zu einer 0 ausgewertetes Literal (sodass in sowohl in S_1 , als auch in S_2 ein Literal enthalten ist), oder es besitzt nur zu 1 ausgewertete Literale (nur zu 0 ausgewertete treten nicht auf, da A erfüllbar ist und wir eine passende Belegung haben). Für den Fall von ausschließlich zu 1 ausgewerteten Literalen gibt es das F , welches als zusätzliches Literal dafür sorgt, dass auch diese rein "positive" Klausel eine gültige Partitionierung erzeugt.

In allen Fällen ist eine gültige Partitionierung möglich.

Aufgabe 2.2.1.d Korrektheit für " \Leftarrow "

Behauptung: Es gibt eine gültige Partitionierung, damit ist A erfüllbar.

Sei S_1, S_2 eine gültige Partitionierung von S mit der Teilmengenmenge C , sowie der dahinter liegenden Formel A , welche wiederum die Literale x_1, x_2, \dots, x_n und die Klauselmengen K besitzt. F ist in der Partition enthalten, in der alle Literale enthalten sind, die in A zu 0 ausgewertet werden sollen. Es kann in dieser Partition nach Definition von SET SPLITTING keine Teilmenge $c \in C$ liegen. In der

anderen Partition sind somit alle Literale enthalten, die zu 1 ausgewertet werden sollen. Wertet man nun jedes Literal aus A zu 1 aus, welches in dieser Partition vorkommt, ist A erfüllt.

Dies ist durch die Konstruktion der Partitionierung gegeben, die besagt, dass Komplemente von Literalen stets in der jeweils anderen Partition enthalten sind. Dadurch wird garantiert, dass es keine Widersprüche in der Auswertung geben kann.

In dieser Partition taucht jedes Literal auf und jedes Literal wird zu 1 ausgewertet, somit ist A mit dieser Belegung erfüllt.

Aufgabe 2.2.2