RWTH Aachen
Lehrgebiet Theoretische Informatik
Kneis—Langer—Richter—Rossmanith

WS 2008/09
Übungsblatt 2
30.10.2008

♡ Übung zur Vorlesung Parametrisierte Algorithmen ♡

Tutoraufgabe T5

Führen Sie den in der Vorlesung besprochenen Algorithmus für Center String auf der folgenden Instanz und für den Parameter $m = 3$ aus.

```
TESTTEST
TESTTEXT
TEXTTEST
FESTTEXT
TEXTFEST
TESTFEST
TTSTFTST
```

Lösungsvorschlag Die Lösung wird live durchgeführt und kann zu Hause auch mit dem Programm auf der Übungsseite im WWW nachvollzogen werden.

Tutoraufgabe T6

Die Größe des Suchbaums im Center String–Algorithmus war gemäß Analyse $(m + 1)^m$ und die Laufzeit daher $O(kn(m + 1)^m)$.
Vereinfachen Sie den Wert für die Laufzeit so weit wie möglich.
Hintergrund: Zu komplizierte Formeln können schnell verwirrend werden; vor allem, wenn zum Beispiel dieser Algorithmus als Unterprogramm in einem komplizierteren Algorithmus verwendet wird, und seine Laufzeit entsprechend eingeht.

Lösungsvorschlag
Uns stört natürlich der lästige Summand 1 in der inneren Klammer. Wir schreiben nun

$$(m + 1)^m = (m \cdot (1 + 1/m))^m = m^m \cdot (1 + 1/m)^m \leq m^m \cdot e.$$

Die hier verwendete Abschätzung $(1 + 1/x)^x \leq e$ ist wohlbekannt, läßt sich aber auch schnell herleiten.

Tutoraufgabe T7

Das Problem MaxSAT ist folgendermaßen definiert:
Eingabe: Eine aussagenlogische Formel F in konjunktiver Normalform
Parameter: k
Frage: Gibt es eine Belegung, die k Klauseln erfüllt?

(a) Finden Sie einen Algorithmus, der MaxSAT in $2^k|F|^{O(1)}$ Schritten löst. Verwenden Sie beschränkte Suchbäume.

(b) Geht es besser als 2^k?

Lösungsvorschlag

Wir können wie folgt verzweigen:

Wir wählen eine Variable. Wenn sie nur positiv oder nur negativ vorkommt, können wir sie entsprechend setzen und erfüllen dadurch mindestens eine Klausel. Ansonsten gibt es zwei Klauseln, so daß die gewählte Variable in der einen Klausel positiv und in der anderen Klausel negativ vorkommt. Dies ergibt offensichtlich den Verzweigungsvektor $(1, 1)$.

Für den zweiten Aufgabenteil läßt sich beobachten, daß der obige Fall durch eine Reduktionsregel erledigt werden kann, wenn die Variable genau einmal positiv und genau einmal negativ vorkommt (Resolution mit Anpassung des Parameters). Somit verbessert sich der Verzweigungsvektor sofort auf $(1, 2)$ und die Laufzeit auf $1.6181^k |F|^{O(1)}$.

Tutoraufgabe T8

Beim in der Vorlesung vorgestellten Algorithmus für Vertex Cover wurden einige Fälle weggelassen. Zum Beispiel dieser: Alle Knoten haben Grad zwischen drei und fünf, kein Knoten mit Grad drei liegt auf einem Dreieck oder hat hat mehr als eine Brücke. Es gibt einen Knoten x mit Grad drei und einer Brücke, und einer seiner Brückennachbarn hat Grad drei.

Wie läßt sich dieser Fall effizient behandeln? Versuchen Sie, auf den Verzweigungsvektor $(3, 3)$ zu kommen.

Lösungsvorschlag

Sei x wie oben, y der Brückenknoten, a, b, c die Nachbarn von x, wobei a der Brückennachbar mit Grad drei sein soll, und d sei der der verbleibende Nachbar von a. Weil x nicht auf einem Dreieck liegt, sind diese Knoten allesamt verschieden.

Wir testen zunächst den Fall, daß x *nicht* im VC enthalten ist. Dieses impliziert, daß a, b und c im VC enthalten sein müssen, was uns zur ersten 3 im Branchingvektor führt.

Schlägt diese Verzweigung fehl, wissen wir, daß x in jeder Lösung enthalten sein muß. Wir zeigen nun, daß es, sollte es überhaupt eine Lösung geben, dann auch immer eine Lösung gibt, die a nicht enthält: Sei C eine beliebige Lösung mit $x \in C$. Wir können ausschließen, daß $y \notin C$, denn das bedeutet $\{a, b\} \subseteq C$, und durch „Schieben" von x auf c erhalten wir wieder obigen Fall, den wir bereits ausgeschlossen haben.

1. Wenn $a \notin C$, ist offensichtlich $\{d, y, x\} \subseteq C$.
2. Wenn $\{a, y, x\} \subseteq C$, dann können wir d statt a wählen und erhalten C' gleicher Größe mit wieder $\{d, y, x\} \subseteq C'$.

Insgesamt finden wir mit den beiden Verzweigungen $\{a, b, c\}$ und $\{d, y, x\}$ immer eine Lösung, sollte es denn eine geben, was uns zum Branchingvektor $(3, 3)$ führt.

Hausaufgabe H3

Entwerfen Sie einen parametrisierten Algorithmus für 3-Hitting Set, das wie folgt definiert ist: Gegeben sind eine Familie $\mathcal{F}$ maximal dreielementiger Mengen über einem gemeinsamen Universum U und eine Zahl $k \in \mathbf{N}$. Zu entscheiden ist, ob es k Elemente aus U gibt, so daß jede Menge aus $\mathcal{F}$ mindestens eines dieser Elemente enthält.

Lösungsvorschlag

Jede Menge von Elementen aus U, die eine Lösung des Problems darstellt, schneidet jede Menge aus $\mathcal{F}$. Mit anderen Worten: Jede Lösung enthält mindestens ein Element jeder Menge aus $\mathcal{F}$. Deshalb können wir wie bei VERTEX COVER (das in der Tat als graphentheoretisches Äquivalent von 2-HITTING SET aufgefasst werden kann) einen Algorithmus entwerfen, der auf einem beschränkten Suchbaum basiert.

Algorithmus *3HS*$(\mathcal{F}, k)$:
if $k < 0$ **return** *false*;
if $\mathcal{F} = \emptyset$ **return** *true*;
let $S = (x_1, x_2, x_3) \in \mathcal{F}$;
if *3HS*$(\{ F \in \mathcal{F} \mid x_1 \notin F \}, k-1)$ **return** *true*;
if *3HS*$(\{ F \in \mathcal{F} \mid x_2 \notin F \}, k-1)$ **return** *true*;
if *3HS*$(\{ F \in \mathcal{F} \mid x_3 \notin F \}, k-1)$ **return** *true*;
return *false*;

Der Rekursionsbaum hat offenkundig $O(3^k)$ Blätter, woraus sich direkt eine Laufzeitschranke von $O(3^k \cdot \mathit{poly}(n))$ ableiten läßt, wenn n die Eingabelänge bezeichnet. Die Korrektheit ist wie beim oben erwähnten Algorithmus für VERTEX COVER leicht zu zeigen.

Hausaufgabe H4

Hier betrachten wir einen weiteren Fall für den in der Vorlesung vorgestellten Algorithmus für VERTEX COVER. Dieser hat unter anderen folgende Voraussetzungen: Alle Knoten haben einen Grad zwischen 3 und 5. Es gibt einen Knoten mit Grad 3, der zwei Nachbarn hat, welche durch eine Kante verbunden sind.

Nennen wir diesen Knoten x und seine Nachbarn a, b, c, und seien a und b verbunden. Ergänzen Sie den Algorithmus der Vorlesung, so daß dieser Fall effizient behandelt wird. Wie lauten Verzweigungsvektor und -zahl?

Lösungsvorschlag

Die erste Möglichkeit ist wie üblich, daß eine Lösung existiert, die x nicht enthält. Dann gehören natürlich alle Nachbarn von x zur Lösung.

Betrachten wir also den Fall, daß x zu jeder Lösung gehören muss. Da es eine Kante zwischen a und b gibt, muss mindestens einer dieser beiden Knoten zur Lösung gehören. Falls x zu jeder Lösung gehört, kann daher c nicht zur Lösung gehören, da sonst a, b, c eine gleich gute Lösung wie x, c, a bzw. x, c, b wäre. Somit muss die Nachbarschaft von c zur Lösung gehören. Da $3 = |N(x)|$ und $3 \leq |N(c)|$ gilt, erhalten wir mindestens den Verzweigungsvektor $(3, 3)$ und die Verzweigungszahl 1.259921.