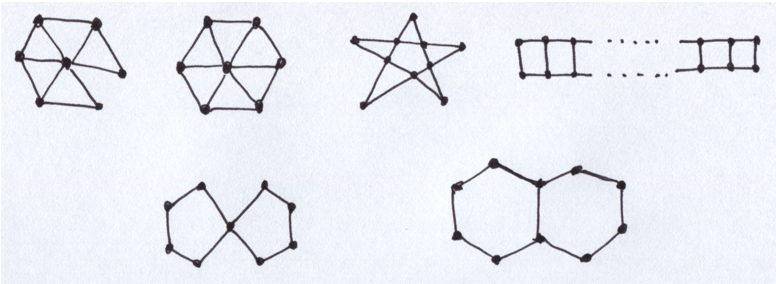
STRUKTURELLE MODELLE IN DER BILDVERARBEITUNG
6. ÜBUNG – PARTIELLE k-BÄUME

Aufgabe 1. Bestimmen Sie die Baumbreite der unten abgebildeten Graphen.



Überlegen Sie für jedes Beispiel, in welcher Reihenfolge die Knoten bei der Dynamischen Programmierung eliminiert werden können. Für welche Teilmengen der Knoten sollen die Bellmannschen Funktionen definiert werden, welche Ordnung haben sie? Schlussfolgern Sie daraus die Zeitkomplexität der Dynamischen Programmierung für jedes Beispiel.

Aufgabe 2. Man betrachte die Aufgabe der Suche nach dem besten aufspannenden Graphen mit bestimmten Eigenschaften. Gegeben sei die Menge R der Knoten. Für jedes Paar $\{r, r'\}$ der Knoten (eine potenzielle Kante) seien Kosten $c(r, r')$ angegeben. Gesucht wird ein aufspannender Graph (R, E), d.h. die Kantenmenge E. Die Kosten einer Kantenmenge ergeben sich als

$$c(E) = \sum_{rr' \in E} c(r, r').$$

Gesucht wird der aufspannende Graph optimaler Kosten.

Betrachte Sie die folgenden Varianten:

a) Gesucht wird eine *aufspannende Kette*, d.h. ein aufspannender Graph, in dem die Kantenmenge einer Kette topologisch äquivalent (isomorph) ist. Mit anderen Worten, wird nach einer vollständigen Ordnung $r_1, r_2 \ldots r_n$ der Menge der Knoten gesucht, die Kanten verbinden die in dieser Ordnung unmittelbar benachbarten Knoten.
Zeigen Sie dass diese Aufgabe NP-vollständig ist.

b) Gesucht wird ein *aufspannender Stern*, d.h. ein aufspannender Baum, in dem alle Knoten bis auf einen Blätter sind.
Geben Sie einen effizienten Algorithmus zur Lösung dieser Aufgabe an.

Aufgabe 3. Man betrachte das Markovsche Modell auf einer Kette und die folgende Modifikation der Aufgabe der Suche nach der a-posteriori wahrscheinlichsten Zustandsfolge. Die Qualität (entspricht dem Logarithmus der Wahrscheinlichkeit) einer Zustandsfolge $y = (y_1, y_2 \dots y_n)$ ergibt sich als

$$Q(y) = \sum_{i=1}^{n} q_i(y_i) + \sum_{i=2}^{n} g_i(y_{i-1}, y_i).$$

Im Gegensatz zum „gewöhnlichen“ Problem der Suche nach der Zustandsfolge maximaler Qualität

$$y^* = \arg\max_y Q(y)$$

wird jetzt nach den m besten Zustandsfolgen gesucht, d.h. nach einem Tupel $(y^1, y^2 \dots y^m)$ aus m Zustandsfolgen so, dass

$$Q(y^1) \geq Q(y^2) \geq \dots \geq Q(y^m) \geq Q(\text{„alle anderen } y\text{“})$$

gilt.

Geben Sie einen effizienten Algorithmus zur Lösung dieses Problems an.