Algorithmen für Routenplanung

Übung 5

Thomas Pajor

Lehrstuhl für Algorithmik I
Institut für theoretische Informatik
Karlsruher Institut für Technologie
Universität Karlsruhe (TH)

6. Juli 2009

Aufgabe 1: SHARC-Routing

Gegeben: Graph $G = (V, E, \text{len})$, eine k-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

Aufgabe 1: SHARC-Routing

Gegeben: Graph $G = (V, E, \text{len})$, eine k-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Wie können bei Kanten $e \in C$ auf Level i, die durch Knotenreduktion entfernt werden, die Arc-Flags auf *allen* Leveln korrekt gesetzt werden?

Aufgabe 1: SHARC-Routing

Gegeben: Graph $G = (V, E, \text{len})$, eine k-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Wie können bei Kanten $e \in C$ auf Level i, die durch Knotenreduktion entfernt werden, die Arc-Flags auf *allen* Leveln korrekt gesetzt werden?

(b) Geben Sie ein effizientes Verfahren zur Verfeinerung der Arc-Flags an.

Aufgabe 1: SHARC-Routing

Gegeben: Graph $G = (V, E, \text{len})$, eine k-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Wie können bei Kanten $e \in C$ auf Level i, die durch Knotenreduktion entfernt werden, die Arc-Flags auf *allen* Leveln korrekt gesetzt werden?

(b) Geben Sie ein effizientes Verfahren zur Verfeinerung der Arc-Flags an.

(c) Zeigen Sie die Korrektheit der Arc-Flags-Kompression wie in der Vorlesung eingeführt. Beschränken Sie sich auf 1-Level-Partitionen.

Aufgabe 2: Transit-Node Routing & Arc-Flags

Gegeben: Graph $G = (V, E, \text{len})$ und Transit-Knoten $\mathcal{T}_1 \subseteq \mathcal{T}_0$ mit $\mathcal{T}_0 = V$ und eine Partitionierung von $\mathcal{T}_1$ in k Zellen. Für jeden Transit-Knoten $a \in \mathcal{T}$ soll ein Bitvektor der Länge k verwendet werden.

Aufgabe 2: Transit-Node Routing & Arc-Flags

Gegeben: Graph $G = (V, E, \text{len})$ und Transit-Knoten $\mathcal{T}_1 \subseteq \mathcal{T}_0$ mit $\mathcal{T}_0 = V$ und eine Partitionierung von $\mathcal{T}_1$ in k Zellen. Für jeden Transit-Knoten $a \in \mathcal{T}$ soll ein Bitvektor der Länge k verwendet werden.

(a) Wann ist zu einer s-t-Anfrage ein Table-Lookup zwischen zwei Access-Nodes $a \in \overrightarrow{A}(s)$ und $a' \in \overleftarrow{A}(t)$ unnötig? Welche Information soll demnach an die Bitvektoren gespeichert werden?

Aufgabe 2: Transit-Node Routing & Arc-Flags

Gegeben: Graph $G = (V, E, \text{len})$ und Transit-Knoten $\mathcal{T}_1 \subseteq \mathcal{T}_0$ mit $\mathcal{T}_0 = V$ und eine Partitionierung von $\mathcal{T}_1$ in k Zellen. Für jeden Transit-Knoten $a \in \mathcal{T}$ soll ein Bitvektor der Länge k verwendet werden.

(a) Wann ist zu einer s-t-Anfrage ein Table-Lookup zwischen zwei Access-Nodes $a \in \overrightarrow{A}(s)$ und $a' \in \overleftarrow{A}(t)$ unnötig? Welche Information soll demnach an die Bitvektoren gespeichert werden?

(b) Geben Sie ein Verfahren zur Berechnung der Bitvektoren an.

Aufgabe 2: Transit-Node Routing & Arc-Flags

Gegeben: Graph $G = (V, E, \text{len})$ und Transit-Knoten $\mathcal{T}_1 \subseteq \mathcal{T}_0$ mit $\mathcal{T}_0 = V$ und eine Partitionierung von $\mathcal{T}_1$ in k Zellen. Für jeden Transit-Knoten $a \in \mathcal{T}$ soll ein Bitvektor der Länge k verwendet werden.

(a) Wann ist zu einer s-t-Anfrage ein Table-Lookup zwischen zwei Access-Nodes $a \in \overrightarrow{A}(s)$ und $a' \in \overleftarrow{A}(t)$ unnötig? Welche Information soll demnach an die Bitvektoren gespeichert werden?

(b) Geben Sie ein Verfahren zur Berechnung der Bitvektoren an.

(c) Wie lässt sich die Transit-Node-Query modifizieren, um von den vorberechneten Zusatzinformationen gebrauch zu machen?