Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Janina Sontheim, Florian Richter

München, 27.06.2019

Algorithmen und Datenstrukturen

SS 2019

Übungsblatt Global 6: Graphen

Aufgabe Global 6-1 *Knobelei: Brussel Sprouts*

Wir definieren das folgende Zwei-Spieler-Spiel: Auf eine Ebene werden zwei Kreuze gesetzt. Abwechselnd führen die beiden Spieler einen Zug durch, wobei stets zwei beliebige freie Enden verbunden werden dürfen. Jede Verbindung muss dabei überschneidungsfrei sein. Nach dem Verbinden wird in der Mitte der Kante ein neuer Strich gesetzt, der zwei neue Enden einführt, und der nächste Spieler ist am Zug. Kann ein Spieler keine gültige Kante mehr einfügen, so hat er verloren. Gibt es eine Gewinnstrategie für einen der Spieler? Terminiert dieses Spiel immer? Was ändert sich, wenn die Anzahl der Startkreuze vergrößert wird?

Lösungsvorschlag:

Beim Ausprobieren stellt man schnell fest: Nach 8 Zügen ist das Spiel immer vorbei und Spieler 1 gewinnt, egal was beide tun. Dies können wir sogar mathematisch beweisen.

Wir modellieren dazu die Kreuze als Knoten eines Graphen und die Verbindungen als Kanten. Zu Beginn starten wir also mit 2 Knoten und 0 Kanten. Einen überschneidungsfreien Graphen nennt man auch planar und er erfüllt stets die Eulercharakteristik #Knoten-#Kanten+#Flächen = 2, wobei zu den Flächen alle durch Kanten eingeschlossenen Flächen sowie die unendliche äußere Fläche zählt.

Zuerst: Warum terminiert das Spiel? Der einzige erste Zug besteht im Verbinden beider Kreuze. Danach werden in jedem Zug für eine gewählte Fläche alle darin befindlichen Enden durch einen neuen Kantenzug separiert. Egal, in welchem Verhältnis dies geschieht, die Fläche wird zu zwei neuen Flächen und beide enthalten jeweils mindestens ein Ende durch das neu eingefügte Kreuz. Da die Anzahl der Enden immer konstant bleibt, muss das Spiel irgendwann terminieren.

Angenommen, das Spiel endet nach m Schritten. Nach den anfänglichen 2 Knoten werden in jedem Schritt 1 Knoten und 2 Kanten hinzugefügt. Also gibt es nach dem $m$ten Zug $m+2$ Knoten und $2m$ Kanten. Da das Spiel endet, befindet sich in jeder Fläche genau ein Ende. Wären dort mehr Enden, wäre das Spiel nicht zuende. Und in jeder Fläche muss es mindestens ein Ende geben, sonst wurde beim Kantenziehen vorher etwas falsch gemacht. Wie zuvor gesagt ist die Anzahl der Enden konstant und genau 8 für zwei Kreuze. In die Eulercharakteristik eingesetzt ergibt dies:

$$v - e + f = m + 2 - 2m + 8 = 2 \Leftrightarrow 8 = m$$

Also endet das Spiel mit zwei Startknoten stets nach genau 8 Zügen. Die Frage nach einer Strategie hat sich damit erübrigt, denn es ist völlig egal, welche Kanten gezogen werden. Bei zwei Startkreuzen gewinnt immer Spieler 1.

Startet man mit n Kreuzen, so gilt

$$v - e + f = m + n - 2m + 4n = 2 \Leftrightarrow 5n - 2 = m$$

Für drei Kreuze endet das Spiel also nach $5 \cdot 3 - 2 = 13$ Zügen. So kann man auch den anderen Spieler wählen lassen, ob er beginnen möchte. Falls er dies bejaht, fügt man noch ein Kreuz hinzu, damit „es interessanter ist“.

Aufgabe Global 6-2 *Flussüberquerung*

Ein Bauer möchte alle seine Waren (Tiere und Gemüse) $x_1, x_2, \ldots, x_n$ zum Markt bringen und muss dazu einen Fluss überqueren. Dazu kann er ein Floß benutzen, auf dem immer nur er und m Objekte Platz haben. In der klassischen Variante besitzt er die drei Objekte Wolf, Salat, Ziege und kann immer nur ein Objekt $m = 1$ überschiffen. Es gibt Paarungen von Waren, die sich niemals gemeinsam auf einer Flussseite befinden dürfen, während der Bauer auf der gegenüberliegenden Seite ist (Wolf frisst Salat). Geben Sie einen Algorithmus an, der dies systematischer löst als Trial-and-Error.

Lösungsvorschlag:

Wir modellieren einen Graphen, der den Zustandsraum für dieses Problem repräsentiert. Der Knoten, der den Start des Problems modelliert, wird mit $(x_1, x_2, \ldots, x_n \leftarrow \_)$ beschriftet. Der Zielknoten ist $(\_ \rightarrow x_1, x_2, \ldots, x_n)$. Der Pfeil markiert die Position, in der sich der Bauer befindet. Knoten werden mit Kanten verbunden, falls der Übergang eine gültige Flussüberquerung darstellt. Knoten, die einen Zustand markieren, der nicht erlaubt ist („Wolf frisst Ziege“), dann bekommt dieser Knoten keine ausgehenden Kanten. Alle Kanten tragen Gewicht 1. Auf diesem Graphen führen wir Dijkstra aus dem Startzustand als Ausgangsknoten und den Endzustand als Ziel. Falls es einen kürzesten Pfad gibt, so ist dies eine gültige Lösung. Den Graph für die klassische Variante geben wir exemplarisch an:

Aufgabe Global 6-3 *Maximale Bipartite Matchings*

Sie stranden auf einer Insel mit $n - 1$ weiteren Personen. Es ist keine Zeit zu verlieren, denn zum Überleben müssen diverse Aufgaben (Wasser, Nahrung, Unterkunft,...) erledigt werden. Dabei haben alle Personen Präferenzen und Fähigkeiten und können daher jeweils nur einen Teil der m Aufgaben erledigen. Entwerfen Sie einen Algorithmus, der möglichst vielen Personen jeweils einen Job zuweist und dabei auch möglichst viele Jobs abdeckt. Hinweis: Berücksichtigen Sie dabei die Algorithmen der Vorlesung.

Lösungsvorschlag:

Wir generieren für alle Personen jeweils einen Knoten $p_1, \ldots, p_n$ und ebenso für alle Jobs die Knoten $j_1, \ldots, j_m$. Wir verbinden Personenknoten mit Jobknoten, falls die Person diesen Job ausführen kann. Alle diese Kanten erhalten außerdem eine Kapazität von 1. Nun führen wir zwei weitere Knoten s, t ein. Für jede Person p führen wir eine Kante (s, p) mit Kapazität 1 ein. Analog fügen wir für jeden Job j die Kante (j, t) mit Kapazität 1 ein. Anschließend bestimmen wir den maximalen Fluss in diesem Netzwerk von s zu t (z.B. Edmonds-Karp). Die im Fluss genutzten Kanten zwischen Personen und Jobs gehören zur optimalen Lösung dieses Matchingproblems.

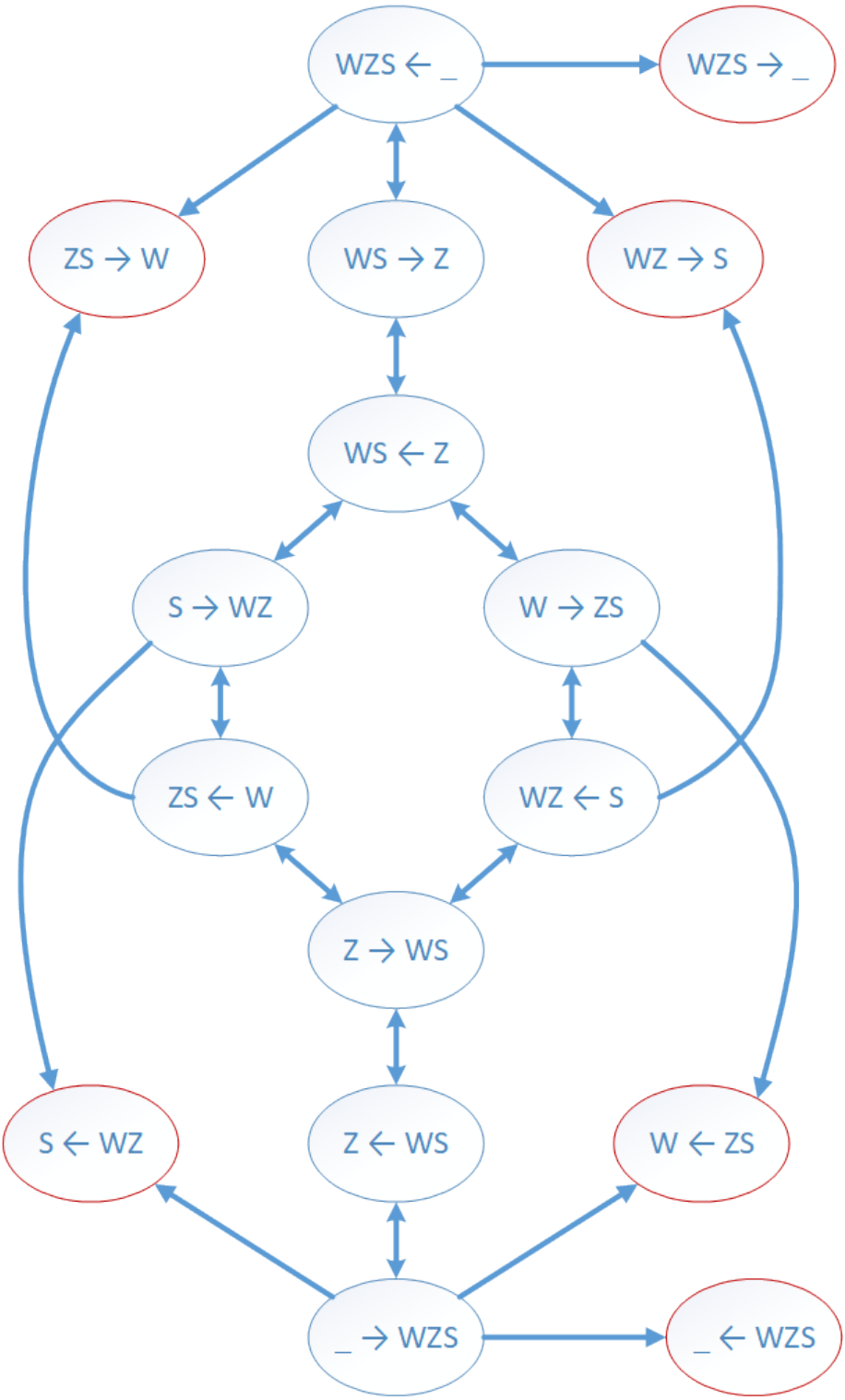

Abbildung 1: 6-2: Graph zur Flussüberquerung

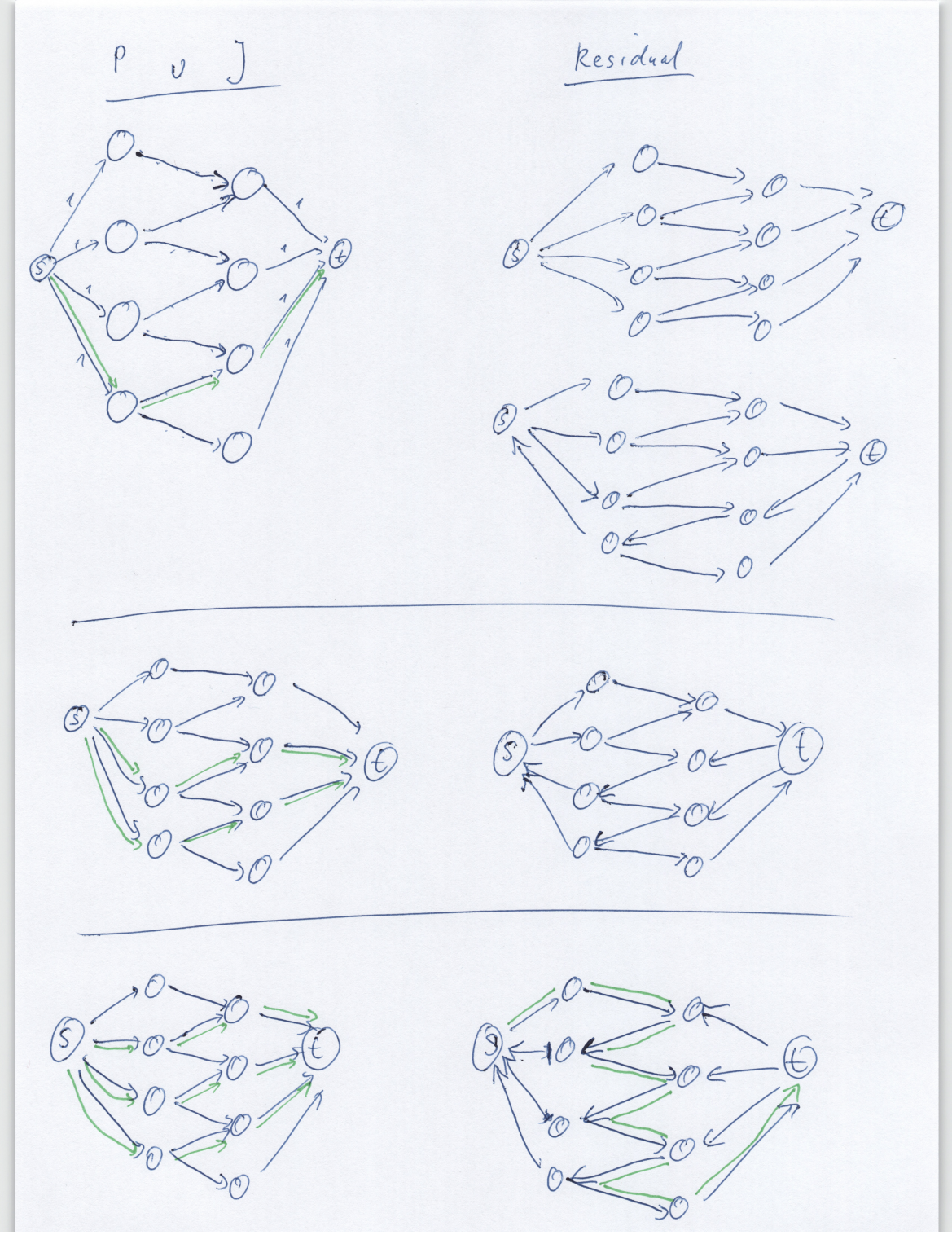

Abbildung 2: 6-3: Beispiel Matching durch Flussnetzwerk