WS 2003/04

Diskrete Strukturen I

Ernst W. Mayr

mayr@in.tum.de
Institut für Informatik
Technische Universität München

01-23-2004

Breitensuche, *breadth first search*

Sei $G = (V, E)$ ein ungerichteter Graph, gegeben mittels Adjazenzlisten.

```
algorithm BFS
  for all v ∈ V do
    touched[v] := false
    bfsNum[v] := 0
  od
  count := 0
  queue := ∅
  for all v ∈ V do
    if not touched[v] then
      bfsLevel[v] := 0
      parent[v] := null
      queue.append(v)
      touched[v] := true
      while not empty(queue) do
        u := remove_first(queue)
        bfsNum[u] := ++count
```

Fortsetzung

```
          for all w ∈ adjacency_list[u] do
            if not touched[w] then
              type[(u,w)] := 'Baumkante'
              parent[w] := u
              bfsLevel[w] := bfsLevel[u]+1
              queue.append(w)
              touched[w] := true
            elsif not w = parent[u] then
              type[(u,w)] := 'Querkante'
            fi
          od
        od
      fi
    od
  end
```

Beobachtungen:

1. Die Breitensuche konstruiert einen Spannwald.
2. Der Spannwald besteht genau aus den Baumkanten im Algorithmus.
3. (u, v) ist Querkante $\Rightarrow \left|\mathrm{bfsLevel}(u) - \mathrm{bfsLevel}(v)\right| \leq 1$

Beobachtungen:

1. Die Breitensuche konstruiert einen Spannwald.
2. Der Spannwald besteht genau aus den Baumkanten im Algorithmus.
3. (u, v) ist Querkante $\Rightarrow \left|\mathrm{bfsLevel}(u) - \mathrm{bfsLevel}(v)\right| \leq 1$

Beobachtungen:

1. Die Breitensuche konstruiert einen Spannwald.
2. Der Spannwald besteht genau aus den Baumkanten im Algorithmus.
3. (u, v) ist Querkante $\Rightarrow \left|\text{bfsLevel}(u) - \text{bfsLevel}(v)\right| \leq 1$

Satz

Der Zeitbedarf für die Breitensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V| + |E|\big)\ .$$

Beweis.

Aus Algorithmus ersichtlich. $\square$

Satz
Der Zeitbedarf für die Breitensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V| + |E|\big) \; .$$

Beweis.
Aus Algorithmus ersichtlich. $\square$

Matroide

Definition

Sei S eine endliche Menge, $U \subseteq 2^S$ eine Teilmenge der Potenzmenge von S. Dann heißt $M = (S, U)$ ein *Matroid und jedes* $A \in U$ *heißt unabhängige Menge, falls gilt:*

1. $\emptyset \in U$
2. $A \in U,\ B \subseteq A \quad \Longrightarrow \quad B \in U$
3. $$A, B \in U,\ |B| = |A| + 1$$
$$\Longrightarrow \quad (\exists x \in B \setminus A)\Big[(A \cup \{x\}) \in U\Big]$$

Jede bezüglich $\subseteq$ maximale Menge in U heißt Basis. Nach 3. haben je zwei Basen gleiche Kardinalität. Diese heißt der Rang $r(M)$ des Matroids.

Matroide

Definition

Sei S eine endliche Menge, $U \subseteq 2^S$ eine Teilmenge der Potenzmenge von S. Dann heißt $M = (S, U)$ ein *Matroid und jedes $A \in U$ heißt unabhängige Menge, falls gilt:*

1. $\emptyset \in U$
2. $A \in U,\ B \subseteq A \quad \Longrightarrow \quad B \in U$
3. $$A, B \in U,\ |B| = |A| + 1$$ $$\Longrightarrow \quad (\exists x \in B \setminus A)\Big[(A \cup \{x\}) \in U\Big]$$

Jede bezüglich $\subseteq$ maximale Menge in U heißt Basis. Nach 3. haben je zwei Basen gleiche Kardinalität. Diese heißt der Rang $r(M)$ des Matroids.

Matroide

Definition

Sei S eine endliche Menge, $U \subseteq 2^S$ eine Teilmenge der Potenzmenge von S. Dann heißt $M = (S, U)$ ein *Matroid und jedes $A \in U$ heißt unabhängige Menge, falls gilt:*

1. $\emptyset \in U$
2. $A \in U,\ B \subseteq A \quad \Longrightarrow \quad B \in U$
3. $$A, B \in U,\ |B| = |A| + 1$$
   $$\Longrightarrow \quad (\exists x \in B \setminus A)\Big[\big(A \cup \{x\}\big) \in U\Big]$$

Jede bezüglich $\subseteq$ maximale Menge in U heißt Basis. Nach 3. haben je zwei Basen gleiche Kardinalität. Diese heißt der Rang r(M) des Matroids.

Matroide

Definition

Sei S eine endliche Menge, $U \subseteq 2^S$ eine Teilmenge der Potenzmenge von S. Dann heißt $M = (S, U)$ ein *Matroid und jedes $A \in U$ heißt unabhängige Menge, falls gilt:*

1. $\emptyset \in U$
2. $A \in U,\ B \subseteq A \quad \Longrightarrow \quad B \in U$
3. $$A, B \in U,\ |B| = |A| + 1 \qquad \Longrightarrow \quad (\exists x \in B \setminus A)\Big[\big(A \cup \{x\}\big) \in U\Big]$$

Jede bezüglich $\subseteq$ maximale Menge in U heißt Basis. Nach 3. haben je zwei Basen gleiche Kardinalität. Diese heißt der Rang r(M) des Matroids.

Matroide

Definition

Sei S eine endliche Menge, $U \subseteq 2^S$ eine Teilmenge der Potenzmenge von S. Dann heißt $M = (S, U)$ ein *Matroid und jedes $A \in U$ heißt unabhängige Menge, falls gilt:*

1. $\emptyset \in U$
2. $A \in U,\ B \subseteq A \quad \Longrightarrow \quad B \in U$
3. $$A, B \in U,\ |B| = |A| + 1$$
$$\Longrightarrow \quad (\exists x \in B \setminus A)\Big[\big(A \cup \{x\}\big) \in U\Big]$$

Jede bezüglich $\subseteq$ maximale Menge in U heißt Basis. Nach 3. haben je zwei Basen gleiche Kardinalität. Diese heißt der Rang r(M) des Matroids.

Beispiel

Linear unabhängige Vektoren in einem Vektorraum.

Beispiel

G sei folgender Graph:

$S =$ Menge der Kanten von G

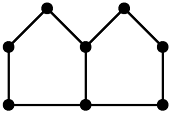
$U =$ Menge der kreisfreien Teilmengen von S

Beispiel

Linear unabhängige Vektoren in einem Vektorraum.

Beispiel

G sei folgender Graph:



$S =$ Menge der Kanten von G

$U =$ Menge der kreisfreien Teilmengen von S

Greedy-Algorithmus

Sei $M = (S, U)$ ein Matroid, $w : S \to R$ eine Gewichtsfunktion.

```
algorithm greedy(S,U,w)
  B := ∅
  while (|B|<r(M)) do
    sei x ∈ {y ∈ S \ B; B ∪ {y} ∈ U} mit minimalem Gewicht
    B := B ∪ {x}
  od
end
```

Satz

Der Greedy-Algorithmus liefert eine Basis minimalen Gewichts.

Satz

Der Greedy-Algorithmus liefert eine Basis minimalen Gewichts.

Beweis

Aus der Definition des Matroids (1.) folgt, dass die leere Menge $\emptyset$ eine unabhängige Menge ist. Aus 3. folgt, dass in der `while`-Schleife wiederum nur unabhängige Mengen generiert werden. Daher ist B am Ende des Algorithmus eine Basis (da inklusionsmaximal). Es bleibt zu zeigen, dass die gefundene Basis minimales Gewicht besitzt. Sei also $B = \{b_1, \ldots, b_r\}$ die vom Algorithmus gelieferte Basis. Wir nehmen nun an, es existiere eine Basis $B' = \{b'_1, \ldots, b'_r\}$ mit $w(B') < w(B)$.

Sei $b_1, \ldots, b_r$ die Reihenfolge der Elemente, in der sie der Greedy-Algorithmus ausgewählt hat. Dann gilt

$$w(b_1) \leq w(b_2) \leq \ldots \leq w(b_r).$$

Beweis

Aus der Definition des Matroids (1.) folgt, dass die leere Menge $\emptyset$ eine unabhängige Menge ist. Aus 3. folgt, dass in der `while`-Schleife wiederum nur unabhängige Mengen generiert werden. Daher ist B am Ende des Algorithmus eine Basis (da inklusionsmaximal). Es bleibt zu zeigen, dass die gefundene Basis minimales Gewicht besitzt. Sei also $B = \{b_1, \ldots, b_r\}$ die vom Algorithmus gelieferte Basis. Wir nehmen nun an, es existiere eine Basis $B' = \{b'_1, \ldots, b'_r\}$ mit $w(B') < w(B)$.

Sei $b_1, \ldots, b_r$ die Reihenfolge der Elemente, in der sie der Greedy-Algorithmus ausgewählt hat. Dann gilt

$$w(b_1) \leq w(b_2) \leq \ldots \leq w(b_r).$$

Beweis

Aus der Definition des Matroids (1.) folgt, dass die leere Menge $\emptyset$ eine unabhängige Menge ist. Aus 3. folgt, dass in der `while`-Schleife wiederum nur unabhängige Mengen generiert werden. Daher ist B am Ende des Algorithmus eine Basis (da inklusionsmaximal). Es bleibt zu zeigen, dass die gefundene Basis minimales Gewicht besitzt. Sei also $B = \{b_1, \ldots, b_r\}$ die vom Algorithmus gelieferte Basis. Wir nehmen nun an, es existiere eine Basis $B' = \{b'_1, \ldots, b'_r\}$ mit $w(B') < w(B)$.

Sei $b_1, \ldots, b_r$ die Reihenfolge der Elemente, in der sie der Greedy-Algorithmus ausgewählt hat. Dann gilt

$$w(b_1) \leq w(b_2) \leq \ldots \leq w(b_r).$$

Beweis

Aus der Definition des Matroids (1.) folgt, dass die leere Menge $\emptyset$ eine unabhängige Menge ist. Aus 3. folgt, dass in der `while`-Schleife wiederum nur unabhängige Mengen generiert werden. Daher ist B am Ende des Algorithmus eine Basis (da inklusionsmaximal). Es bleibt zu zeigen, dass die gefundene Basis minimales Gewicht besitzt. Sei also $B = \{b_1, \ldots, b_r\}$ die vom Algorithmus gelieferte Basis. Wir nehmen nun an, es existiere eine Basis $B' = \{b'_1, \ldots, b'_r\}$ mit $w(B') < w(B)$.

Sei $b_1, \ldots, b_r$ die Reihenfolge der Elemente, in der sie der Greedy-Algorithmus ausgewählt hat. Dann gilt

$$w(b_1) \leq w(b_2) \leq \ldots \leq w(b_r).$$

O. B. d. A. gelte auch

$$w(b'_1) \leq w(b'_2) \leq \ldots \leq w(b'_r) \; .$$

Da nach Annahme $w(B') < w(B)$ gilt, muss es ein minimales i geben mit $w(b'_i) < w(b_i)$. Annahme: Es gibt ein (kleinstes) $k \in \{1, \ldots, i-1\}$, so dass $b_k \notin \{b'_1, \ldots, b'_{i-1}\}$. Dann können wir b_k gegen ein geeignetes $b'_{k'}$ austauschen, das wegen der Optimalität von B' und der Unabhängigkeit von $\{b_1, \ldots, b_k\}$ gleiches Gewicht wie b_k haben muss. Also können wir o.B.d.A. annehmen, dass $\{b_1, \ldots, b_{i-1}\} = \{b'_1, \ldots, b'_{i-1}\}$.

O. B. d. A. gelte auch

$$w(b'_1) \le w(b'_2) \le \ldots \le w(b'_r) \ .$$

Da nach Annahme $w(B') < w(B)$ gilt, muss es ein minimales i geben mit $w(b'_i) < w(b_i)$. Annahme: Es gibt ein (kleinstes) $k \in \{1, \ldots, i-1\}$, so dass $b_k \notin \{b'_1, \ldots, b'_{i-1}\}$. Dann können wir b_k gegen ein geeignetes $b'_{k'}$ austauschen, das wegen der Optimalität von B' und der Unabhängigkeit von $\{b_1, \ldots, b_k\}$ gleiches Gewicht wie b_k haben muss. Also können wir o.B.d.A. annehmen, dass $\{b_1, \ldots, b_{i-1}\} = \{b'_1, \ldots, b'_{i-1}\}$.

O. B. d. A. gelte auch

$$w(b'_1) \leq w(b'_2) \leq \ldots \leq w(b'_r) \,.$$

Da nach Annahme $w(B') < w(B)$ gilt, muss es ein minimales i geben mit $w(b'_i) < w(b_i)$. Annahme: Es gibt ein (kleinstes) $k \in \{1, \ldots, i-1\}$, so dass $b_k \notin \{b'_1, \ldots, b'_{i-1}\}$. Dann können wir b_k gegen ein geeignetes $b'_{k'}$ austauschen, das wegen der Optimalität von B' und der Unabhängigkeit von $\{b_1, \ldots, b_k\}$ gleiches Gewicht wie b_k haben muss. Also können wir o.B.d.A. annehmen, dass $\{b_1, \ldots, b_{i-1}\} = \{b'_1, \ldots, b'_{i-1}\}$.

Nach 3. gibt es ein $b'_j \in \{b'_1, \ldots, b'_i\} \setminus \{b_1, \ldots, b_{i-1}\}$ so, dass $\{b_1, \ldots, b_{i-1}, b'_j\}$ unabhängig ist. Aber dann gilt

$$w(b'_j) \leq w(b'_i) < w(b_i)$$

im Widerspruch zur minimalen Auswahl von b_i im Algorithmus.
q. e. d.

Nach 3. gibt es ein $b'_j \in \{b'_1, \dots, b'_i\} \setminus \{b_1, \dots, b_{i-1}\}$ so, dass $\{b_1, \dots, b_{i-1}, b'_j\}$ unabhängig ist. Aber dann gilt

$$w(b'_j) \leq w(b'_i) < w(b_i)$$

im Widerspruch zur minimalen Auswahl von b_i im Algorithmus.
q. e. d.

Minimale Spannbäume

Satz

Sei $G = (V, E)$ ein zusammenhängender, ungerichteter Graph, $F \subseteq 2^E$ die Menge der kreisfreien Teilmengen von E. Dann ist $M = (E, F)$ ein Matroid mit Rang $|V| - 1$.

Beweis

Es sind die drei Eigenschaften eines Matroids zu zeigen.

1. $\emptyset$ ist kreisfrei und daher in F enthalten.
2. Ist A kreisfrei und B eine Teilmenge von A, dann ist auch B kreisfrei.

Satz

Sei $G = (V, E)$ ein zusammenhängender, ungerichteter Graph, $F \subseteq 2^E$ die Menge der kreisfreien Teilmengen von E. Dann ist $M = (E, F)$ ein Matroid mit Rang $|V| - 1$.

Beweis

Es sind die drei Eigenschaften eines Matroids zu zeigen.

1. $\emptyset$ ist kreisfrei und daher in F enthalten.
2. Ist A kreisfrei und B eine Teilmenge von A, dann ist auch B kreisfrei.

Minimale Spannbäume

Satz

Sei $G = (V, E)$ *ein zusammenhängender, ungerichteter Graph,* $F \subseteq 2^E$ *die Menge der kreisfreien Teilmengen von* E*. Dann ist* $M = (E, F)$ *ein Matroid mit Rang* $|V| - 1$*.*

Beweis

Es sind die drei Eigenschaften eines Matroids zu zeigen.

1. $\emptyset$ ist kreisfrei und daher in F enthalten.
2. Ist A kreisfrei und B eine Teilmenge von A, dann ist auch B kreisfrei.

1. Sind A und B kreisfrei, $|B| = |A| + 1$, dann existiert ein $b \in B$, so dass $A \cup \{b\}$ kreisfrei ist: Wir betrachten die Wälder (V, A) (mit $|A|$ Kanten und $|V| - |A|$ Zusammenhangskomponenten) und (V, B) (mit $|B|$ Kanten und $|V| - |B|$ Zusammenhangskomponenten). Diese Bedingungen lassen zwei Möglichkeiten zu:
   1. Es existiert eine Kante e in B, die zwei Zusammenhangskomponenten in (V, A) verbindet. Damit ist $A \cup \{e\}$ kreisfrei.
   2. Alle Kanten in B verlaufen innerhalb der Zusammenhangskomponenten in (V, A). (V, A) besitzt jedoch eine Zusammenhangskomponente mehr als (V, B). Daher muss es eine Zusammenhangskomponente in (V, A) geben, deren Knoten nicht in (V, B) auftauchen, was einen Widerspruch darstellt.

q. e. d.

1. Sind A und B kreisfrei, $|B| = |A| + 1$, dann existiert ein $b \in B$, so dass $A \cup \{b\}$ kreisfrei ist: Wir betrachten die Wälder (V, A) (mit $|A|$ Kanten und $|V| - |A|$ Zusammenhangskomponenten) und (V, B) (mit $|B|$ Kanten und $|V| - |B|$ Zusammenhangskomponenten). Diese Bedingungen lassen zwei Möglichkeiten zu:
   1. Es existiert eine Kante e in B, die zwei Zusammenhangskomponenten in (V, A) verbindet. Damit ist $A \cup \{e\}$ kreisfrei.
   2. Alle Kanten in B verlaufen innerhalb der Zusammenhangskomponenten in (V, A). (V, A) besitzt jedoch eine Zusammenhangskomponente mehr als (V, B). Daher muss es eine Zusammenhangskomponente in (V, A) geben, deren Knoten nicht in (V, B) auftauchen, was einen Widerspruch darstellt.

q. e. d.

1. Sind A und B kreisfrei, $|B| = |A| + 1$, dann existiert ein $b \in B$, so dass $A \cup \{b\}$ kreisfrei ist: Wir betrachten die Wälder (V, A) (mit $|A|$ Kanten und $|V| - |A|$ Zusammenhangskomponenten) und (V, B) (mit $|B|$ Kanten und $|V| - |B|$ Zusammenhangskomponenten). Diese Bedingungen lassen zwei Möglichkeiten zu:
   1. Es existiert eine Kante e in B, die zwei Zusammenhangskomponenten in (V, A) verbindet. Damit ist $A \cup \{e\}$ kreisfrei.
   2. Alle Kanten in B verlaufen innerhalb der Zusammenhangskomponenten in (V, A). (V, A) besitzt jedoch eine Zusammenhangskomponente mehr als (V, B). Daher muss es eine Zusammenhangskomponente in (V, A) geben, deren Knoten nicht in (V, B) auftauchen, was einen Widerspruch darstellt.

q. e. d.