Vorlesung Diskrete Strukturen
Eulersche und Hamiltonsche Graphen

Bernhard Ganter

Institut für Algebra
TU Dresden
D-01062 Dresden
bernhard.ganter@tu-dresden.de

WS 2009/10

Kantenzug

Ein **Kantenzug** in einem Graphen (V, E) ist eine Folge

$$(a_0, a_1, \ldots, a_n)$$

von Ecken mit der Eigenschaft, dass für alle $i \in \{0, 1, \ldots, n-1\}$ die aufeinanderfolgenden Ecken $\{a_i, a_{i+1}\}$ eine Kante bilden. Ein Kantenzug $(a_0, \ldots, a_n)$ heißt **offen**, falls $a_0 \neq a_n$ ist und anderenfalls **geschlossen**.

(Im Unterschied zum **Weg** müssen die Ecken in einem Kantenzug nicht paarweise verschieden sein.)

Kantenzug

Ein **Kantenzug** in einem Graphen (V, E) ist eine Folge

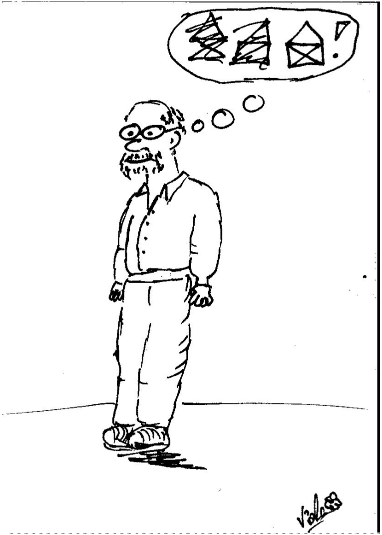
$$(a_0, a_1, \ldots, a_n)$$

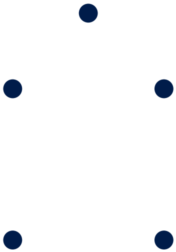
von Ecken mit der Eigenschaft, dass für alle $i \in \{0, 1, \ldots, n-1\}$ die aufeinanderfolgenden Ecken $\{a_i, a_{i+1}\}$ eine Kante bilden. Ein Kantenzug $(a_0, \ldots, a_n)$ heißt **offen**, falls $a_0 \neq a_n$ ist und anderenfalls **geschlossen**.

(Im Unterschied zum **Weg** müssen die Ecken in einem Kantenzug nicht paarweise verschieden sein.)

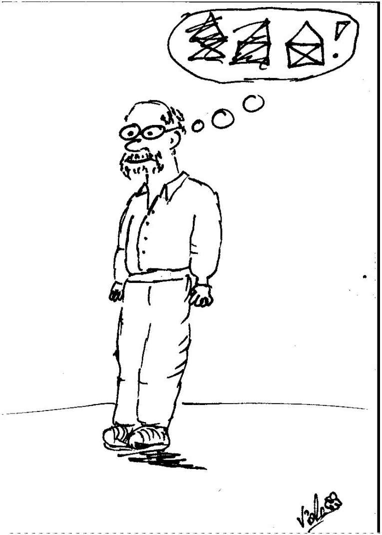
Ein Kantenzug, in dem jede Kante des Graphen genau einmal auftritt und der alle Ecken durchläuft, wird eine **Eulersche Linie** genannt.

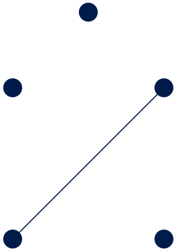
Das Haus vom Nikolaus



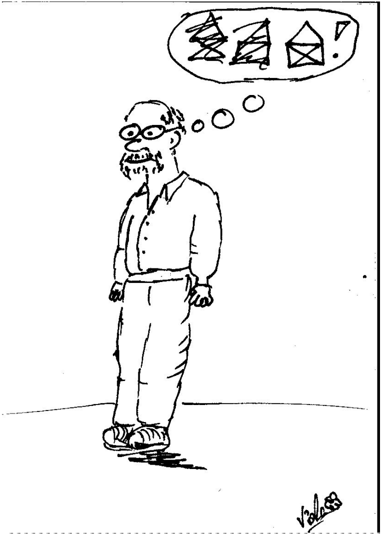


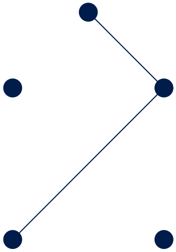
Das Haus vom Nikolaus



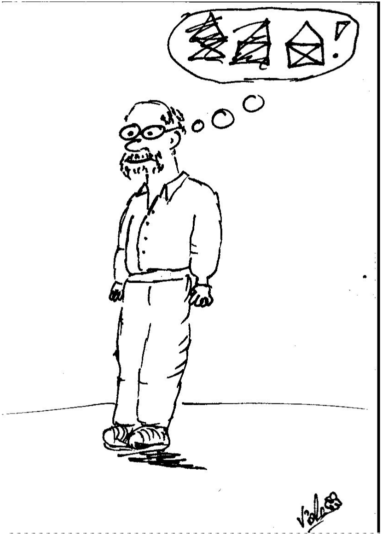


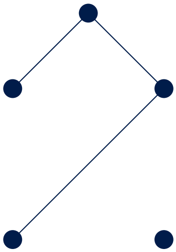
Das Haus vom Nikolaus



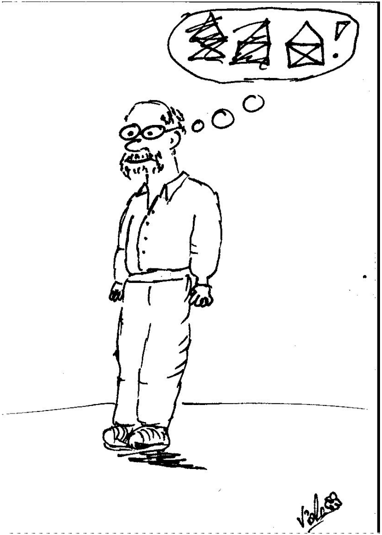


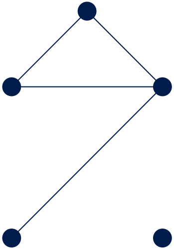
Das Haus vom Nikolaus



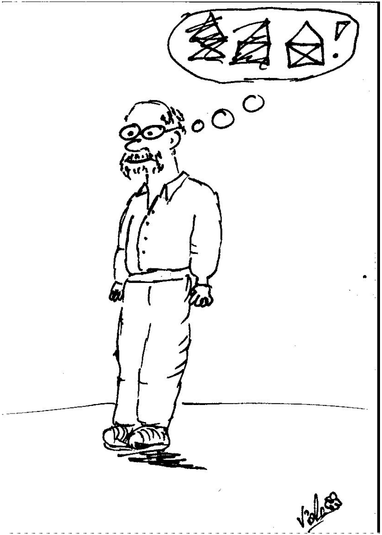


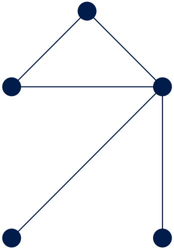
Das Haus vom Nikolaus



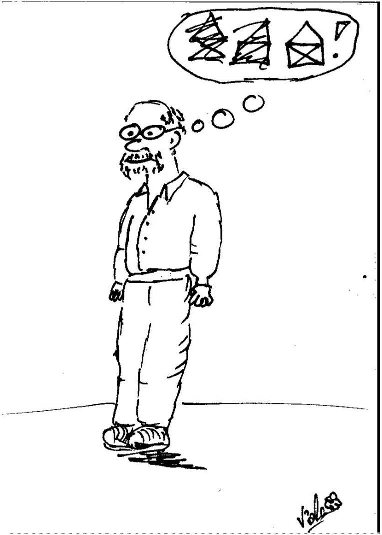


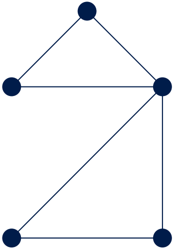
Das Haus vom Nikolaus



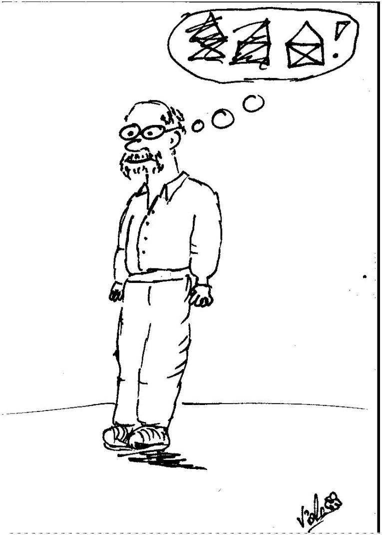


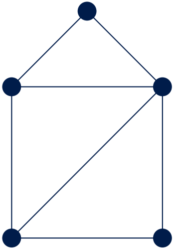
Das Haus vom Nikolaus



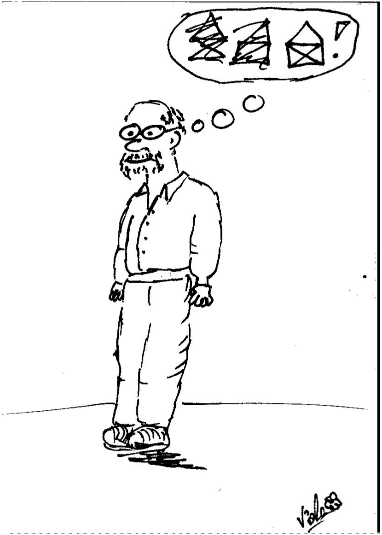


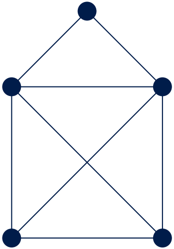
Das Haus vom Nikolaus





Das Haus vom Nikolaus





Eulersche Linien

Welche Figuren lassen sich in einem Zug zeichnen, d.h. ohne den Stift abzusetzen?

Diese Frage kann in die Sprache der Graphendiagramme übersetzt werden und lautet dann:

Eulersche Linien

Welche Figuren lassen sich in einem Zug zeichnen, d.h. ohne den Stift abzusetzen?

Diese Frage kann in die Sprache der Graphendiagramme übersetzt werden und lautet dann:

Welche Graphen besitzen eine Eulersche Linie?

Eulersche Linien

Welche Figuren lassen sich in einem Zug zeichnen, d.h. ohne den Stift abzusetzen?

Diese Frage kann in die Sprache der Graphendiagramme übersetzt werden und lautet dann:

Welche Graphen besitzen eine Eulersche Linie?

Diese Frage wurde als das *Königsberger Brückenproblem* bekannt und 1736 von Leonhard Euler beantwortet.

Eulers Ergebnis

Satz (Euler 1736)

Ein endlicher Graph besitzt genau dann eine geschlossene Eulersche Linie, wenn er zusammenhängend ist und jede seiner Ecken geraden Grad hat.

Ein endlicher Graph besitzt genau dann eine offene Eulersche Linie, wenn er zusammenhängend ist und genau zwei Ecken ungeraden Grades hat.

Eulers Ergebnis

Satz (Euler 1736)

Ein endlicher Graph besitzt genau dann eine geschlossene Eulersche Linie, wenn er zusammenhängend ist und jede seiner Ecken geraden Grad hat.

Ein endlicher Graph besitzt genau dann eine offene Eulersche Linie, wenn er zusammenhängend ist und genau zwei Ecken ungeraden Grades hat.

Zu diesem Satz geben wir einen *Beweis* und einen *Algorithmus* an.

Beweisvorbereitung

Notation: Ist $Z := (a_0, a_1, \ldots, a_n)$ ein Kantenzug, dann soll $V(Z)$ die Menge der in Z auftretenden Ecken und $E(Z)$ die Menge der in Z auftretenden Kanten bezeichnen.

Beweisvorbereitung

Notation: Ist $Z := (a_0, a_1, \dots, a_n)$ ein Kantenzug, dann soll $V(Z)$ die Menge der in Z auftretenden Ecken und $E(Z)$ die Menge der in Z auftretenden Kanten bezeichnen.

Vorüberlegung: In einer früheren Vorlesung war gezeigt worden, dass jeder nichtleere kreislose zusammenhängende endliche Graph eine Ecke vom Grad ≤ 1 enthält. Daraus schließen wir folgendes:

Ein zusammenhängender endlicher Graph (V, E) mit $E \neq \emptyset$, in dem alle Eckengrade grade sind, enthält einen Kreis und damit einen nichtleeren geschlossenen Kantenzug, der jede Kante höchstens einmal durchläuft.

Beweisteil 1

Zu beweisen ist: *Ein Graph, der eine geschlossene Eulersche Linie besitzt, ist zusammenhängend und alle seine Ecken haben geraden Eckengrad.*

Beweis: Weil vorausgesetzt wurde, dass eine Eulersche Linie alle Ecken durchläuft, muss ein solcher Graph zusammenhängend sein. Eine beliebige Ecke $v \in V$ des Graphen darf mehrfach in der Eulerschen Linie $(a_n, \ldots, a_n)$ auftreten. Für jede Zahl $i < n$ mit $v = a_i$ sind $\{a_{i-1}, v\}$ und $\{v, a_{i+1}\}$ (Indizes $\bmod n$) Kanten durch v, und alle diese Kanten sind verschieden. Der Eckengrad von v erhöht sich also für jedes solche Auftreten um 2 und ist deshalb gerade.

Beweisteil 2

Zu beweisen ist: *Ein zusammenhängender endlicher Graph, in dem alle Ecken geraden Grad haben, besitzt eine geschlossene Eulersche Linie.*

Beweis: Wir dürfen $E \neq \emptyset$ annehmen. Nach der Vorüberlegung entält (V, E) dann einen Kreis und damit einen nichtleeren geschlossenen Kantenzug, der jede Kante höchstens einmal durchläuft. Sei Z ein solcher Kantenzug maximaler Kantenanzahl. Wenn Z alle Kanten des Graphen enthält, sind wir fertig.

Beweisteil 2, Fortsetzung

Wenn der maximale Kantenzug Z nicht alle Kanten enthält, dann gibt es eine Kante, die nicht zu Z gehört. Weil (V, E) zusammenhängend ist, muss es sogar eine solche Kante geben, die mit dem Kantenzug eine Ecke gemeinsam hat, sagen wir u.

Der Graph $(V, E \setminus E(Z))$, also der Graph, der entsteht, wenn man den Kantenzug Z in (V, E) löscht, hat ebenfalls nur Ecken von geradem Grad. Deshalb gibt es einen geschlossenen Kantenzug durch u in $(V, E \setminus E(Z))$, der jede Kante höchstens einmal enthält. Dieser Kantenzug kann beim Durchlaufen der Ecke u in Z eingeschoben werden, im Widerspruch zur Maximalität von Z.

Beweisteil 3: Offene Eulersche Linien

Ist (V, E) ein endlicher zusammenhängender Graph mit genau zwei Ecken u, v von ungeradem Grad, und ist $w \notin V$, dann ist $(V \cup \{w\}, E \cup \{\{u, w\}, \{w, v\}\})$ ein endlicher zusammenhängender Graph, in dem alle Ecken geraden Grad haben. Nach dem, was wir bereits bewiesen haben, muss dieser Graph eine geschlossene Eulersche Linie besitzen. Streicht man darin die Ecke w und die beiden Kanten durch w, so erhält man eine offene Eulersche Linie in (V, E).

Umgekehrt sind die genannten Bedingungen (endlich, zusammenhängend, genau zwei Ecken von ungeradem Grad) offenbar notwendig für die Existenz eine offenen Eulerschen Linie.

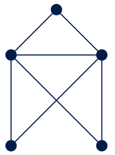
Quod erat demonstrandum.

Existenz und Konstruktion

Wir haben (u.a.) einen **Existenzbeweis** geführt, der nachweist, dass unter den genannten Bedingungen eine Eulersche Linie existiert.

Ein solcher Beweis ist nicht notwendig **konstruktiv**, d.h., er gibt keinen Algorithmus an, eine Eulersche Linie tatsächlich zu konstruieren.

Wie findet man eine Eulersche Linie? Einfach losmarschieren führt nicht sicher zum Ziel, wie nebenstehendes Beispiel zeigt. Alle Voraussetzungen sind erfüllt: Der Graph ist zusammenhängend, endlich und alle Ecken haben geraden Grad.

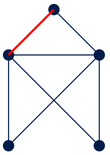


Existenz und Konstruktion

Wir haben (u.a.) einen **Existenzbeweis** geführt, der nachweist, dass unter den genannten Bedingungen eine Eulersche Linie existiert.

Ein solcher Beweis ist nicht notwendig **konstruktiv**, d.h., er gibt keinen Algorithmus an, eine Eulersche Linie tatsächlich zu konstruieren.

Wie findet man eine Eulersche Linie? Einfach losmarschieren führt nicht sicher zum Ziel, wie nebenstehendes Beispiel zeigt. Alle Voraussetzungen sind erfüllt: Der Graph ist zusammenhängend, endlich und alle Ecken haben geraden Grad.

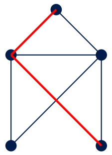


Existenz und Konstruktion

Wir haben (u.a.) einen **Existenzbeweis** geführt, der nachweist, dass unter den genannten Bedingungen eine Eulersche Linie existiert.

Ein solcher Beweis ist nicht notwendig **konstruktiv**, d.h., er gibt keinen Algorithmus an, eine Eulersche Linie tatsächlich zu konstruieren.

Wie findet man eine Eulersche Linie? Einfach losmarschieren führt nicht sicher zum Ziel, wie nebenstehendes Beispiel zeigt. Alle Voraussetzungen sind erfüllt: Der Graph ist zusammenhängend, endlich und alle Ecken haben geraden Grad.

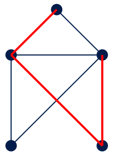


Existenz und Konstruktion

Wir haben (u.a.) einen **Existenzbeweis** geführt, der nachweist, dass unter den genannten Bedingungen eine Eulersche Linie existiert.

Ein solcher Beweis ist nicht notwendig **konstruktiv**, d.h., er gibt keinen Algorithmus an, eine Eulersche Linie tatsächlich zu konstruieren.

Wie findet man eine Eulersche Linie? Einfach losmarschieren führt nicht sicher zum Ziel, wie nebenstehendes Beispiel zeigt. Alle Voraussetzungen sind erfüllt: Der Graph ist zusammenhängend, endlich und alle Ecken haben geraden Grad.

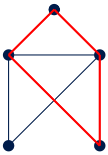


Existenz und Konstruktion

Wir haben (u.a.) einen **Existenzbeweis** geführt, der nachweist, dass unter den genannten Bedingungen eine Eulersche Linie existiert.

Ein solcher Beweis ist nicht notwendig **konstruktiv**, d.h., er gibt keinen Algorithmus an, eine Eulersche Linie tatsächlich zu konstruieren.

Wie findet man eine Eulersche Linie? Einfach losmarschieren führt nicht sicher zum Ziel, wie nebenstehendes Beispiel zeigt. Alle Voraussetzungen sind erfüllt: Der Graph ist zusammenhängend, endlich und alle Ecken haben geraden Grad.



Ein Algorithmus zur Konstruktion einer Eulerschen Linie

Der folgende Algorithmus konstruiert zu jedem endlichen zusammenhängenden Graphen (V, E), in dem alle Eckengrade gerade sind, eine geschlossene Eulersche Linie.
Unter einer **Brücke** in einem Graphen versteht man eine Kante, deren Wegnahme die Anzahl der Zusammenhangskomponenten erhöht.

1. Starte in einer beliebigen Ecke a_0, setze $i := 0$ und $Z_0 := (a_0)$.
2. Ist $Z_i := (a_0, \ldots, a_i)$ bereits konstruiert, verfahre wie folgt:
    - Wenn Z_i alle Kanten des Graphen durchläuft, stopp.
    - Ansonsten wähle eine Kante $\{a_i, w\} \notin E(Z_i)$.
      Wenn es möglich ist, wähle diese Kante so, dass sie keine Brücke im Graphen $(V, E \setminus E(Z_i))$ ist.
    - Setze $i \leftarrow i + 1$, $a_i := w$ und $Z_i := (a_0, \ldots, a_i)$.
3. Iteriere (2).

Funktioniert das?

Es ist gar nicht klar, dass der angegebene Algorithmus zum gewünschten Ergebnis führt.

Ja, es ist nicht einmal klar, dass er überhaupt durchführbar ist.

Betrachten wir z.B. Schritt (2). Dort heißt es:

- Wenn Z_i alle Kanten des Graphen durchläuft, stopp.
- Ansonsten wähle eine Kante $\{a_i, w\} \notin E(Z_i)$ ...

Aber wieso muss es eine solche Kante überhaupt geben?

Funktioniert das?

Es ist gar nicht klar, dass der angegebene Algorithmus zum gewünschten Ergebnis führt.

Ja, es ist nicht einmal klar, dass er überhaupt durchführbar ist.

Betrachten wir z.B. Schritt (2). Dort heißt es:

- Wenn Z_i alle Kanten des Graphen durchläuft, stopp.
- Ansonsten wähle eine Kante $\{a_i, w\} \notin E(Z_i)$ ...

Aber wieso muss es eine solche Kante überhaupt geben?

Offenbar ist es erforderlich, einen sorgfältigen **Korrektheitsbeweis** für diesen Algorithmus anzugeben.

Korrektheitsbeweis (1)

Zu beweisen ist, dass der Algorithmus durchführbar ist und zu einer Eulerschen Linie führt.

Offensichtlich ist, dass für alle i die konstruierte Folge Z_i ein Kantenzug ist, in dem wegen $\{a_i, w\} \notin E(Z_i)$ keine Kante zweimal vorkommt.

Da der Algorithmus erst stoppt, wenn Z_i *alle* Kanten von (V, E) durchläuft, muss das Ergebnis eine (geschlossene) Eulersche Linie sein.

Zu beweisen ist also nur, dass der Algorithmus durchführbar ist, also dass immer dann, wenn der konstruierte Kantenzug $Z_i = (a_0, \ldots, a_i)$ nicht alle Kanten durchläuft, noch eine Kante $\{a_i, w\} \notin E(Z_i)$ existiert.

Korrektheitsbeweis (2)

Nehmen wir also an, der Kantenzug $Z_i = (a_0, \ldots, a_i)$ sei nach den Regeln des Algorithmus konstruiert, es sei $E(Z_i) \neq E$, aber es gäbe keine Kante $\{a_i, w\} \notin E(Z_i)$.

Korrektheitsbeweis (2)

Nehmen wir also an, der Kantenzug $Z_i = (a_0, \ldots, a_i)$ sei nach den Regeln des Algorithmus konstruiert, es sei $E(Z_i) \neq E$, aber es gäbe keine Kante $\{a_i, w\} \notin E(Z_i)$.

Weil a_i geraden Eckengrad hat, muss es auch gerade viele Kanten in $E(Z_i)$ geben, die a_i enthalten. Das erzwingt $a_i = a_0$.

Korrektheitsbeweis (2)

Nehmen wir also an, der Kantenzug $Z_i = (a_0, \ldots, a_i)$ sei nach den Regeln des Algorithmus konstruiert, es sei $E(Z_i) \neq E$, aber es gäbe keine Kante $\{a_i, w\} \notin E(Z_i)$.

Weil a_i geraden Eckengrad hat, muss es auch gerade viele Kanten in $E(Z_i)$ geben, die a_i enthalten. Das erzwingt $a_i = a_0$.

a_0 ist deshalb eine Ecke, die in keiner Kante aus $E \setminus E(Z_i)$ vorkommt.

Korrektheitsbeweis (2)

Nehmen wir also an, der Kantenzug $Z_i = (a_0, \ldots, a_i)$ sei nach den Regeln des Algorithmus konstruiert, es sei $E(Z_i) \neq E$, aber es gäbe keine Kante $\{a_i, w\} \notin E(Z_i)$.

Weil a_i geraden Eckengrad hat, muss es auch gerade viele Kanten in $E(Z_i)$ geben, die a_i enthalten. Das erzwingt $a_i = a_0$.

a_0 ist deshalb eine Ecke, die in keiner Kante aus $E \setminus E(Z_i)$ vorkommt.

Es kann nicht sein, dass *alle* a_j, $0 \leq j \leq i$, diese Eigenschaft haben. Sonst wäre nämlich $\{a_0, \ldots, a_i\}$ eine nicht triviale Zusammenhangskomponente des Graphen (V, E).

Korrektheitsbeweis (2)

Nehmen wir also an, der Kantenzug $Z_i = (a_0, \ldots, a_i)$ sei nach den Regeln des Algorithmus konstruiert, es sei $E(Z_i) \neq E$, aber es gäbe keine Kante $\{a_i, w\} \notin E(Z_i)$.

Weil a_i geraden Eckengrad hat, muss es auch gerade viele Kanten in $E(Z_i)$ geben, die a_i enthalten. Das erzwingt $a_i = a_0$.

a_0 ist deshalb eine Ecke, die in keiner Kante aus $E \setminus E(Z_i)$ vorkommt.

Es kann nicht sein, dass *alle* a_j, $0 \leq j \leq i$, diese Eigenschaft haben. Sonst wäre nämlich $\{a_0, \ldots, a_i\}$ eine nicht triviale Zusammenhangskomponente des Graphen (V, E).

Es muss also eine Kante $\{a_j, b\} \in E \setminus E(Z_i)$ geben.