# Triangulierung eines planaren Graphen

Thomas Pajor*

1. Februar 2007

Das Triangulieren eines Graphen ist eine Grundoperation, die von vielen Algorithmen, die auf planaren Graphen operieren, benötigt wird. Der hier vorgestellte Algorithmus trianguliert einen Graphen mit $n$ Knoten in $\mathcal{O}(n)$ Zeit durch geeignetes Einfügen von Kanten.

## Definitionen

Sei $G = (V, E)$ ein einfacher, planarer und zusammenhängender Graph ohne Multikanten und Schlingen. Der triangulierte Graph $G'$ zu $G$ entsteht durch sukzessives Einfügen von Kanten, ohne dabei die Planaritätseigenschaft des Graphen zu verletzen. Am Ende ist jede Facette von $G'$ ein Dreieck, und es gilt für die Anzahl der Knoten $n$ und Kanten $m$ die Beziehung $m = 3n - 6$.

Sei $v$ ein Knoten in $G$ und $e = (v, u)$ eine zu $v$ inzidente Kante. Wir bezeichnen mit $\Theta(e)$ die gegen den Uhrzeigersinn nächste inzidente Kante an $v$, und nennen sie die *Nachfolgekante* von $e$ an $v$. Mit $\Theta^{-1}(e)$ bezeichnen wir die *Vorgängerkante* von $e$ an $v$. Sie ist also die im Uhrzeigersinn nächste Kante an $v$. Außerdem definieren wir mit $\Theta^*(e)$ die gegen den Uhrzeigersinn nächste Kante am Knoten $u$. Siehe auch Abbildung 1 zur Verdeutlichung.

## Der Algorithmus

Die grundlegende Idee des Algorithmus besteht darin, sukzessive jede Kante $e$ des Graphen zu betrachten, und gegebenfalls Kanten so einzufügen, dass nach Betrachtung der Kante $e$ die Facette rechts von $e$ ein Dreieck ist. Dazu untersuchen wir die Kanten $e, \Theta^{-1}(e), \Theta^*(e)$ und $\Theta^*(\Theta^*(e))$ und unterscheiden drei Fälle je nach Konfiguration der Kanten. Wir definieren uns

*E-Mail: pajor@ira.uka.de; Universität Karlsruhe (TH), Forschungsuniversität · gegründet 1825, Fakultät für Informatik, Institut für Theoretische Informatik

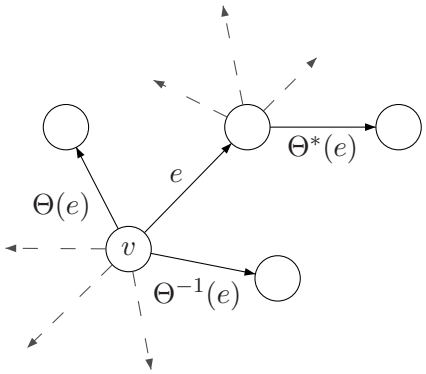



**Abbildung 1:** Definitionen von $\Theta(e)$, $\Theta^{-1}(e)$ und $\Theta^*(e)$.

dazu:

$$
\begin{aligned}
e &:= (v, u) \\
e' &:= (u, u') = \Theta^*(e) \\
e'' &:= (u', u'') = \Theta^*(e') = \Theta^*(\Theta^*(e)) \\
e^- &:= (v, u''') = \Theta^{-1}(e)
\end{aligned}
$$

Für die Abbildungen im Folgenden soll gelten: Gestrichelte dünne Kanten deuten an, dass in der (kombinatorischen) Einbettung des Graphen an dieser Stelle weitere ausgehende Kanten existieren könnten. Die dicke gestrichelte Kante ist jeweils die einzufügende Kante. Ist der Graph ungerichtet, so orientieren wir die Kanten automatisch beim Betrachten des Knotens $v$ und einer zu $v$ inzidenten Kante $e$ auf die Weise, wie in den Abbildungen dargestellt.

## Fall I

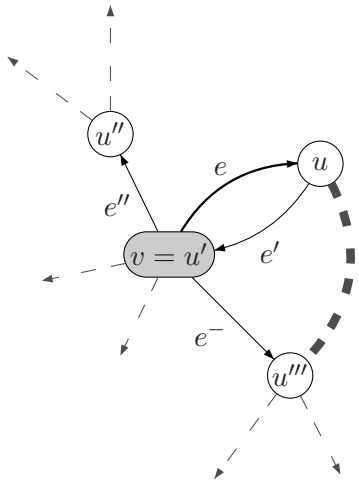
Betrachte Abbildung 2.

Der Knoten $u$ ist nur zur $v$ adjazent (Es gilt also $e = e'$). Damit folgt, dass $u' = v$ gilt, und es genügt eine neue Kante zwischen $u$ und $u'''$ einzufügen. Diese kann kreuzungsfrei eingebettet werden, und es entsteht auch keine Mehrfachkante, da $u$ nur zu $v$ adjazent ist.

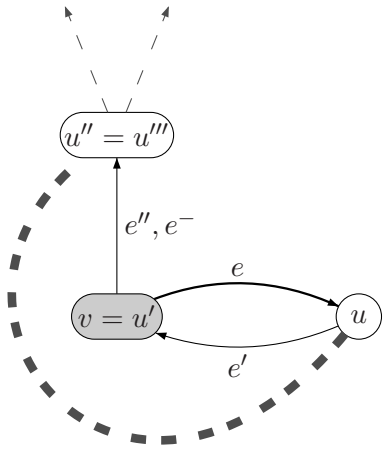
## Fall II

Angenommen $u$ ist zu mindestens einem weiteren Knoten adjazent. Es existiert also eine Kante $e'$ die nicht mit $e$ zusammenfällt. Ist nun jedoch $u'' = v$, so bilden die Kanten $e, e'$ und $e''$ ein Dreieck, und wir brauchen keine Kante einzufügen.

Nehmen wir also an, es gilt $v \neq u'$ und $v \neq u''$. Betrachte dazu auch Abbildung 3.

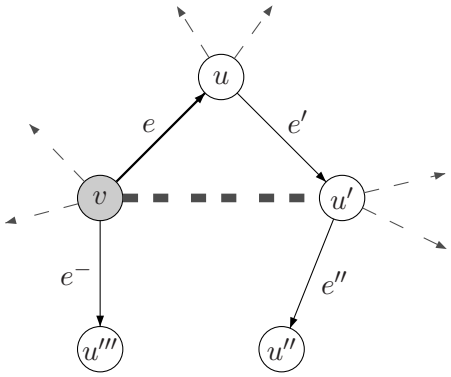



**(a)** Allgemeiner Fall

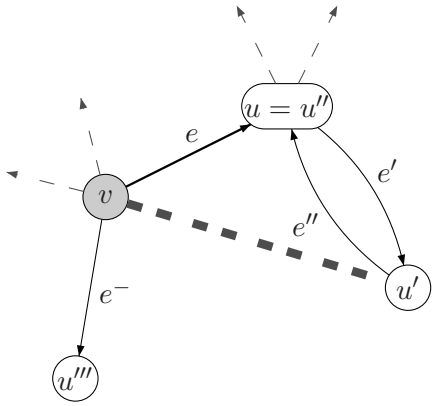



**(b)** Der Knoten $v$ hat nur zwei ausgehende Kanten

**Abbildung 2:** Fall I




**(a)** Allgemeiner Fall

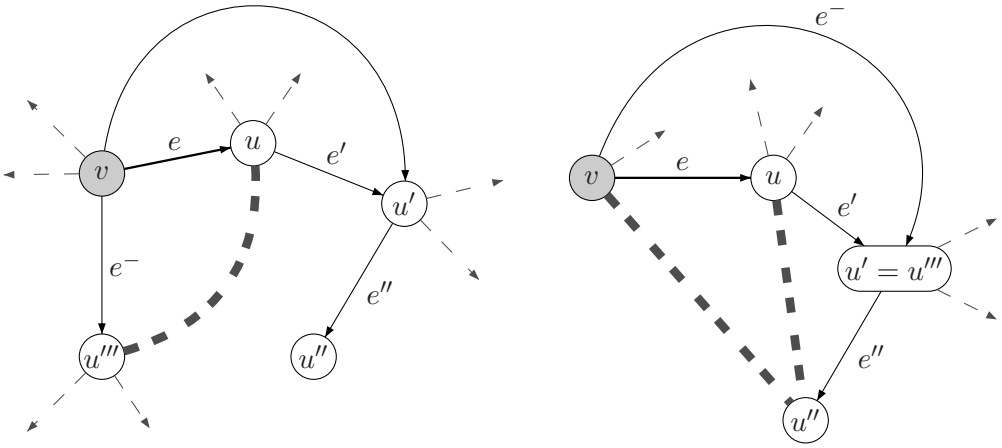



**(b)** Knoten $u'$ ist nur zu $u$ adjazent

**Abbildung 3:** Fall II

Es genügt nun eine Kante zwischen $v$ und $u'$ einzufügen, so dass $\Theta(v, u') = e^-$ und $\Theta(u', v) = e'$. Dann bilden die Kanten $e, e'$ zusammen mit der eingefügten Kante das gesuchte Dreieck.

Es ist aber möglich, dass eine Kante zwischen $v$ und $u'$ bereits existiert, nämlich dadurch, dass sie die Facette links von $e$ begrenzt. Dann würde durch Einfügen einer weiteren Kante zwischen $v$ und $u'$ eine Mehrfachkante entstehen. Dazu untersuchen wir Fall III.




**(a)** Allgemeiner Fall **(b)** Knoten $u''$ ist nur zu $u'$ adjazent

**Abbildung 4:** Fall III

## Fall III

Sei also wieder $v \neq u'$ und $v \neq u''$, aber die Kante $(v, u')$ sei bereits wie in Abbildung 4 im Graphen enthalten, das heißt sie begrenzt die von $e$ aus linke Facette.

Ist $u' \neq u'''$, das heißt $e^- \neq (v, u')$, so können wir eine Kante zwischen $u$ und $u'''$ einfügen, und erhalten so das gesuchte Dreieck. Problematisch ist der Fall, der in Abbildung 4b dargestellt ist, nämlich dass die Kante zwischen $v$ und $u'$ gerade die Kante $e^-$ ist. Damit fallen $u'$ und $u'''$ zusammen, und das Einfügen einer Kante zwischen $u$ und $u'''$ würde eine Mehrfachkante induzieren.

Da $u \neq u''$ gilt, können wir eine Kante zwischen $u$ und $u''$ einfügen, ohne eine Mehrfachkante zu erzeugen. Damit wir rechts von $e$ schließlich ein Dreieck erhalten, fügen wir eine weitere Kante zwischen $v$ und $u''$ ein.

Damit haben wir alle Möglichkeiten betrachtet, wie die Kanten $e, e', ''$ und $e^-$ zueinander liegen können. Der Gesamtalgorithmus iteriert nun über alle Knoten und für jeden der betrachteten Knoten werden die inzidenten Kanten im Uhrzeigersinn abgearbeitet. Algorithmus 1 zeigt eine Implementierung in Pseudo-Code. Eine Bemerkung zur Notation: Ist $v$ ein Knoten und $e = (v, u)$ eine zu $v$ inzidente Kante, so bezeichne $\overline{e}$ die Rückwärtskante $(u, v)$.

**Korrektheit:** Da in jedem Schritt des Algorithmus höchstens eine zu $v$ inzidente Kante eingefügt wird, und diese Kante immer rechts von $e$ eingefügt wird, folgt, dass jede Kante des Graphen genau einmal betrachtet wird. Da nach jedem Einfügeschritt die Facette rechts der betrachteten Kante $e$ ein Dreieck bildet, gilt nach Abarbeitung eines Knotens $v$, dass jede zu $v$ inzidente Facette ein Dreieck bildet. Da wir alle Knoten sukzessive bearbeiten, ist am Ende jede Facette ein Dreieck, und da wir alle neuen Kanten kreuzungsfrei eingebettet haben ohne dabei

Mehrfachkanten zu erzeugen, ist der Graph schließlich trianguliert. □

**Laufzeit:** Jede Kante wird genau einmal betrachtet, da sie, nachdem sie bearbeitet wurde markiert wird, und markierte Kanten nicht mehr bearbeitet werden. Das Setzen des Zeitstempels der Nachbarknoten von $v$ in Zeile 5 kostet genau so viel Schritte wie $v$ inzidente Kanten hat, das heißt, die Zuweisung in Zeile 6 geschieht insgesamt genauso oft wie Kanten betrachtet werden. Die Abfrage $T(u')$ in den Zeilen 15 und 18 kann mit einer geeigneten Datenstruktur in konstanter Zeit realisiert werden. Damit ist die Laufzeit linear in der Anzahl der Kanten des Graphen, und da bei einem planaren Graphen $m \in \mathcal{O}(n)$ gilt, folgt für die Laufzeit des Algorithmus $\mathcal{O}(n)$.

**Algorithmus 1** : TRIANGULIERUNG

**Eingabe** : Planarer Graph $G = (V, E)$
**Seiteneffekte** : $G$ ist trianguliert

**für alle** $v \in V$ **tue**
  $T(v) \leftarrow 0$
$t \leftarrow 0$
**für alle** $v \in V$ **tue**
  $t \leftarrow t + 1$
  **für alle** $w$ *ist Nachbar von* $v$ **tue**
    $T(u) \leftarrow t$
  $e = (v, u) \leftarrow$ Eine aus $v$ ausgehende Kante
  **solange** $e$ *ist nicht markiert* **tue**
    $e' = (u, u') \leftarrow \Theta^*(e)$
    $e'' = (u', u'') \leftarrow \Theta^*(e')$
    $e^- = (v, u''') \leftarrow \Theta^{-1}(e)$
    `// Fall I`
    **wenn** $u' = v$ **dann**
      Füge $e_{\text{neu}} := (u, u''')$ mit $\Theta^{-1}(e_{\text{neu}}) = e'$ ein
      Füge $e_{\text{neu}} := (u''', u)$ mit $\Theta(e_{\text{neu}}) = \overline{e^-}$ ein
    `// Fall II: Die Kante` $(v, u')$ `existiert noch nicht`
    **sonst wenn** $u' \neq v$, $u'' \neq v$ *und* $T(u') < t$ **dann**
      Füge $e_{\text{neu}} := (v, u')$ mit $\Theta(e_{\text{neu}}) = e$ ein
      Füge $e_{\text{neu}} := (u', v)$ mit $\Theta^{-1}(e_{\text{neu}}) = \overline{e'}$ ein
    `// Fall III: Die Kante` $(v, u')$ `existiert bereits`
    **sonst wenn** $u' \neq v$, $u'' \neq v$ *und* $T(u') = t$ **dann**
      **wenn** $u'' \neq u'$ **dann**
        Füge $e_{\text{neu}} := (u, u''')$ mit $\Theta(e_{\text{neu}}) = e'$ ein
        Füge $e_{\text{neu}} := (u''', u)$ mit $\Theta(e_{\text{neu}}) = \overline{e^-}$ ein
      **sonst**
        Füge $e_{\text{neu}} := (u, u'')$ mit $\Theta^{-1}(e_{\text{neu}}) = \overline{e}$ ein
        Füge $e_{\text{neu}} := (u'', u)$ mit $\Theta^{-1}(e_{\text{neu}}) = \overline{e''}$ ein
        Füge $e'_{\text{neu}} := (v, u'')$ mit $\Theta(e'_{\text{neu}}) = e$ ein
        Füge $e'_{\text{neu}} := (u'', v)$ mit $\Theta^{-1}(e'_{\text{neu}}) = e_{\text{neu}}$ ein
    Markiere $e$
    $e \leftarrow \Theta^{-1}(e)$