Triangulierung eines planaren Graphen

Idee Wir möchten Schritt für Schritt alle (gerichteten) Kanten des Graphen betrachten und gegebenenfalls Kanten so einfügen, dass folgende Punkte gelten:

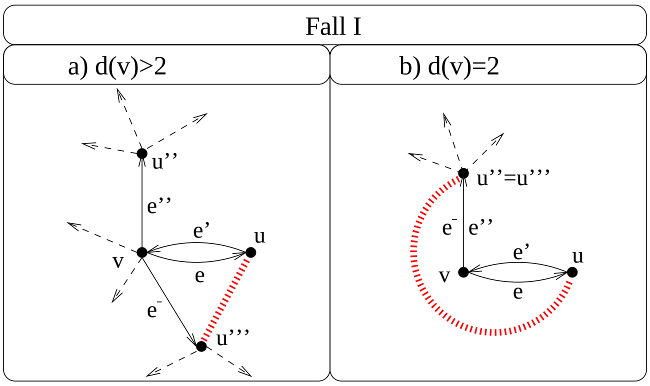
1. Nach Bearbeiten einer Kante $e = (v, u)$ ist die Facette ,,rechts" von e ein Dreieck, begrenzt von den Kanten e, $\Theta^*(e)$, $\Theta^*(\Theta^*(e))$.

2. Beim Bearbeiten der Kante $e = (v, u)$ wird höchstens eine aus v ausgehende Kante (v, v') eingefügt. Für diese Kante gilt dann $\Theta^{-1}(e) = (v, v')$. Weitere eingefügte Kanten sind nicht zu v inzident.

3. Alle eingefügten Kanten können kreuzungsfrei eingebettet werden, es entstehen weder Mehrfachkanten noch Schlingen.

Dazu untersuchen wir alle möglichen Konfigurationen, die die Kanten e, e', e'', e^+, e^- einnehmen können. Diese Kanten sind im Folgenden immer als dünne durchgezogene Linien dargestellt. Andere bestehende Kanten sind als dünne gestrichelte Linien angedeutet, und einzufügende Kanten sind dick gepunktet.

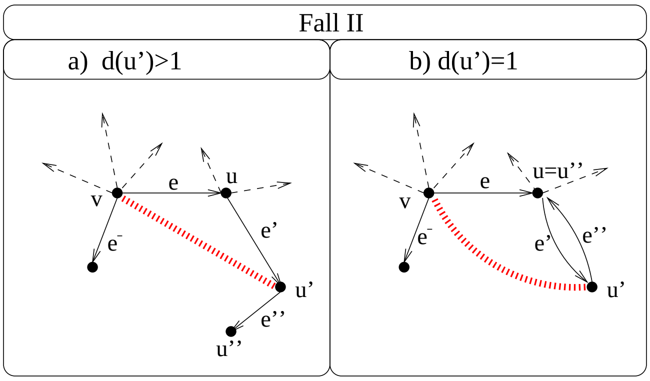
Zuerst wollen wir den Fall I untersuchen, dass der Knoten u nur zu v adjazent ist.



Dann ist $u' = v$, und da $|V| \geq 3$ und G zusammenhängend ist, hat v mindestens zwei ausgehende Kanten. Links ist der Fall dargestellt, dass v mehr als zwei ausgehende Kanten hat, und rechts der Fall von genau zwei ausgehenden Kanten. In beiden Fällen kann (u, u''') bzw. (u''', u) kreuzungsfrei eingebettet werden. Da u zu keinen weiteren Knoten als v adjazent ist, entsteht dadurch auch keine Mehrfachkante, und der Fall I ist abgeschlossen.

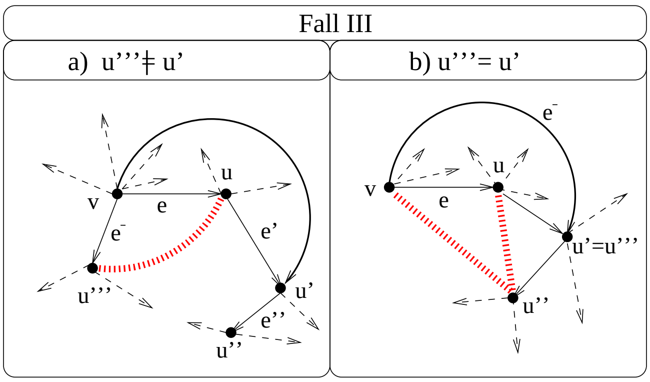
Falls u nicht nur zu v adjazent ist, gilt $u' \neq v$, und falls $u'' = u$, so bilden die Kanten e, e', und e'' bereits ein Dreieck. In diesem Fall wird keine Kante eingefügt, und die obigen Punkte sind bereits erfüllt.

Gilt jedoch $u' \neq v$ und $u'' \neq u$, so ist Punkt 1 für die Kante e nicht erfüllt, und es müssen Kanten eingefügt werden. Dazu würde es prinzipiell genügen, Kanten zwischen v und u' einzufügen.



Wenn diese Kanten so eingefügt werden, dass $\Theta^*(e') = (u', v)$, so bilden e, e', und (u', v) das gewünschte Dreieck.

Problematisch ist nur, dass durch Einfügen dieser Kante eine Mehrfachkanten entstehen könnte, wenn die Kante (v, u') bereits existiert. Diesen Fall wollen wir nun getrennt untersuchen (Fall III).



Hier können wir die Kante (v, u') also nicht so einfügen, dass wir das in 1. gewünschte Dreiech bekommen, es können aber stattdessen andere Kanten eingefügt werden: Falls die Kante $e^- = (v, u''')$, also die im Uhrzeigersinn nach e nächste von v ausgehende Kante, nicht die Kante (v, u') ist, können die Kanten (u, u''') und (u''', u) eingefügt werden. Dies kann kreuzungsfrei gemacht werden, da e bei v direkt nach e^- eingebettet ist. Ausserdem kann vor dem Einfügen keine Kante (u, u''') existieren, da diese die Kante (v, u') kreuzen müsste.

Falls die Kante e^- aber die Kante (v, u') ist, so können folgende Kanten eingefügt werden: Die Kanten (u, u''), (u'', u), (v, u''), und (u'', v) können mit dem Argument wie oben kreuzungsfrei eingebettet werden, und es kann auch hier keine dieser Kanten vorher existieren, da diese wieder (v, u') kreuzen müssten.

Algorithmus

1. Verwalte für jeden Knoten $v \in V$ einen Zeitstempel $t(v) \leftarrow 0$
2. $t \leftarrow 0$
3. Für alle Knoten $v \in V$ führe aus
4. $\quad t \leftarrow t + 1$
5. $\quad$ Für alle Nachbarn u von v setze $t(u) \leftarrow t$
6. $\quad e = (v, u) \leftarrow$ eine aus v ausgehende Kante
7. $\quad$ Solange e nicht markiert
8. $$\begin{aligned} e' = (u, u') &\leftarrow \Theta^*(e) \\ e'' = (u', u'') &\leftarrow \Theta^*(e') \\ e^+ &\leftarrow \Theta(e) \\ e^- = (v, u''') &\leftarrow \Theta^{-1}(e) \end{aligned}$$
9. Falls
   I. $u' = v$:
   füge (u, u''') mit $\Theta(u, u''') = e'$
   und (u''', u) mit $\Theta(u''', u) = e^-$ ein.
   II. $u' \neq v$, $u'' \neq v$, und $t(u') < t$ (d.h. Kante (v, u') existiert nicht):
   füge (v, u') mit $\Theta(v, u') = e$
   und (u', v) mit $\Theta(u', v) = e''$ ein.
   III. $u' \neq v$, $u'' \neq v$, und $t(u') = t$ (d.h. Kante (v, u') existiert):
   a) $u''' \neq u'$:
   füge (u, u''') mit $\Theta(u, u''') = e'$
   und (u''', u) mit $\Theta(u''', u) = e^-$ ein
   b) $u''' = u'$:
   füge (u, u'') mit $\Theta(u, u'') = e'$
   und (u'', u) mit $\Theta^{-1}(u, u'') = \bar{e}''$
   und (v, u'') mit $\Theta(v, u'') = e$
   und (u'', v) mit $\Theta^{-1}(u'', v) = (u'', u)$ ein
10. $\quad$ Markiere e
11. $\quad$ Setze $e \leftarrow \Theta^{-1}(e)$
12. Markiere v

Korrektheit Die Korrektheit des Algorithmus folgt aus der Reihenfolge, in der die Kanten betrachtet werden. Wird ein Knoten v bearbeitet, so werden nacheinander alle ausgehenden Kanten betrachtet, im Uhrzeigersinn beginnend bei einer beliebigen Kante. Werden dabei Kanten eingefügt, so wird aber wegen Punkt 2 nur *eine aus v ausgehende* Kante eingefügt, und diese wird dann sofort im nächsten Schritt bearbeitet. Wenn v markiert wird, sind alle ausgehenden Kanten bearbeitet worden, und wegen Punkt 1 ist dann jede zu v benachbarte Facette ein Dreieck, und es können im Folgenden keine Kanten zu v hinzugefügt werden. Da jeder Knoten markiert wird, ist jeder Knoten nur zu Dreiecken benachbart, und da laut Punkt 3 alle Kanten kreuzungsfrei eingebettet werden und keine Mehrfachkanten entstehen, ist der Graph am Ende des Algorithmus trianguliert.

Laufzeit Nachdem eine Kante bearbeitet wurde, ist sie markiert und wird danach nicht mehr betrachtet, also wird jede Kante im Algorithmus nur einmal bearbeitet. In Schritt 5 wird einmal für jeden Knoten für alle Nachbarn der Zeitstempel gesetzt, also insgesamt auch so oft wie es gerichtete Kanten gibt. Der Test in Schritt 9, ob es eine Kante (v, u') gibt, ist mit der Zeitstempel-Konstruktion dann in konstanter Zeit machbar mittels der Abfrage $t(u') < t$.

Damit ist die Laufzeit insgesamt linear in der Anzahl der Kanten des Graphen, und wegen der Planarität des Graphen also linear in der Anzahl der Knoten.