Satz 2.9

(i) Das Verfahren 2.7 stoppt immer in endlicher Zeit, ist also tatsächlich ein Algorithmus.

(ii) Der Algorithmus liefert einen Weg.

(iii) Ist v_0 ungerade, stoppt der Algorithmus im zweiten ungeraden Knoten.

(iv) Ist G eulersch (d.h. haben alle Knoten geraden Grad), stoppt der Algorithmus in v_0, liefert also einen geschlossenen Weg.

Beweis:

(i) Bei jedem Durchlaufen der Schleifen 2.1-2.5 wird in 2.3 eine Kante entfernt. Dies kann nur endlich oft passieren.

(ii) Nach Konstruktion erhalten wir eine Kantenfolge; keine Kante wird doppelt verwendet.

(iii) Ein gerader Knoten wird genauso oft betreten wie verlassen, also kann der Algorithmus weder im Startknoten noch in einem geraden Knoten stoppen, wenn der Startknoten ungerade ist. Es bleibt nur der andere ungerade Knoten.

(iv) Wie in (iii) kann der Algorithmus in keinem geraden Knoten stoppen, der nicht Anfangsknoten ist; also bleibt nur der Startknoten.

$\square$

Beobachtungen:

Schlecht: Es können bei STOP noch unbenutzte Kanten übrig bleiben.

Gut:

SATZ 2.10

Wenn Algorithmus 2.7 stoppt, bleibt ein eulerscher Graph zurück, d.h. ein Graph mit lauter geraden Knoten.

Beweis:

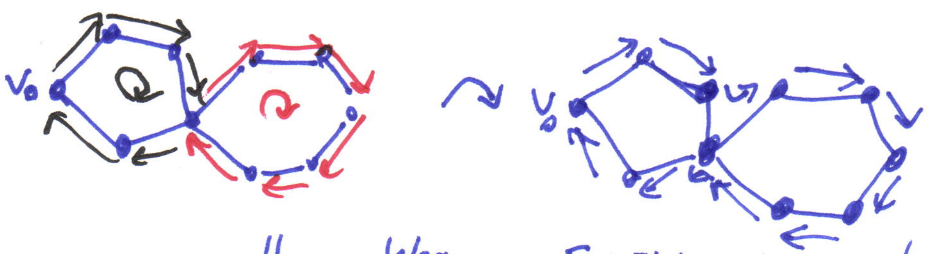
Durch Entfernen des Weges wird für jeden geraden Knoten der Grad um eine gerade Zahl geändert, bleibt also gerade. Für einen der ggf. vorhandenen ungeraden Knoten wird der Graph um eine ungerade Zahl geändert, wird also gerade.

$\square$

Also: Wähle einen neuen Knoten mit positivem Grad, durchlaufe Algorithmus 2.7, verschmelze Wege → Algorithmus 2.8 !

Beobachtung 2.11

(i) Zwei geschlossene Wege mit einem gemeinsamen Knoten kann man in einen geschlossenen Weg verwandeln:



(ii) Man kann aus allen Wegen EINEN Weg machen, wenn der Graph zusammenhängend ist.
(Immer wieder (i) anwenden!)