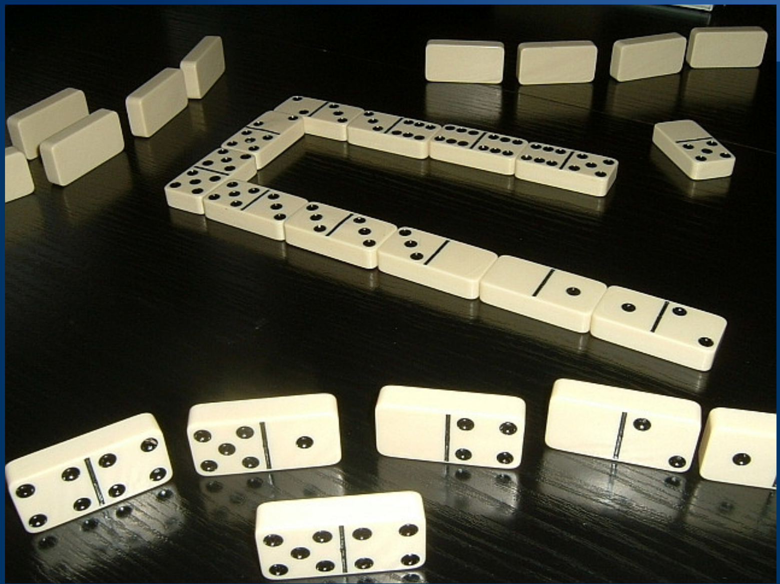
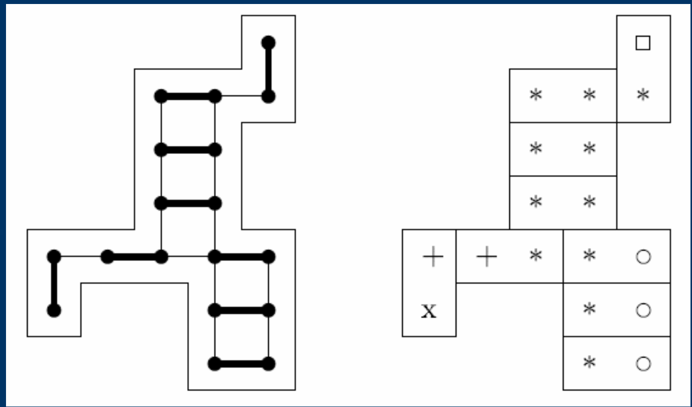
Die Komplexität von Domino

- oder die Frage nach dem Problem, ob man ein Polygon mit einer Menge von Dominosteinen bedecken kann



Referent: Thorsten Reinhardt



Freie Universität Berlin / FB Informatik

Seminar über Algorithmen, SoSe 2007

Dozent: Prof. Dr. Günter Rote

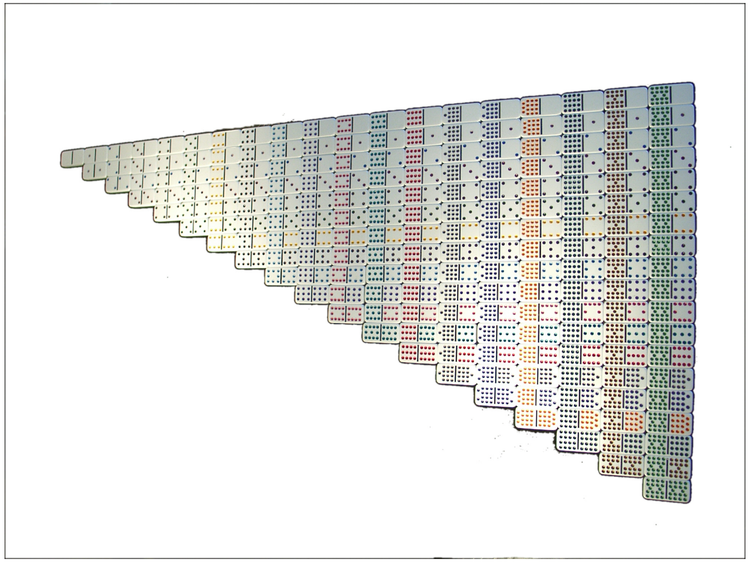
Freie Universität Berlin

Gliederung

1. Einführung
2. Vorbereitungen
3. Domino Tiling ist NP-schwer
4. Zusammenfassung
5. Literatur- und Quellenangaben

Domino – das Spiel

- Legespiel mit rechteckigen (2 x 1) Spielsteinen
- Steine sind in zwei Hälften geteilt
- Grundregel: abwechselnd Steine mit Feldern mit gleicher Augenzahl aneinanderfügen
- wer zuerst alle Steine angelegt hat, ist Sieger



Domino Tiling (grob: Pflasterung, Zerlegung)

Inwieweit lässt sich ein gegebenes Polygon (ganzzahlige Seitenlänge, Seiten stehen jeweils senkrecht aufeinander) mit Hilfe von Dominos unter Berücksichtigung der Grundregel komplett abgedecken, ohne dass weder etwas über das Polygon herausragt, noch dass Stellen frei bleiben.

Domino Tiling ist NP-vollständig

- Domino Tiling ist NP-vollständig ist, selbst mit nur 3 verwendeten Farben
- Umwandlung des Problems ineine Graphenstruktur umgewandelt
- Abbilden auf einen Graphen, welcher 3-färbbar ist.

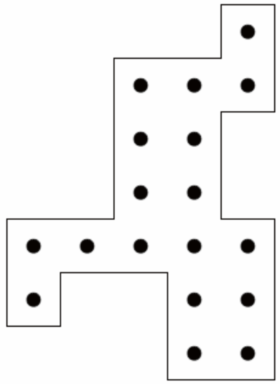
(Das Problem, zu entscheiden, ob ein Graph 3-färbbar ist, ist bekanntermaßen NP-schwer.)

- Beschränkung auf Zeigen der NP-Schwere
- Die NP-Vollständigkeit (Frage, ob Domino Tiling in NP) ergibt sich auf recht einfache Weise:

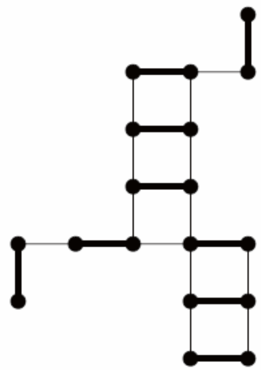
Es gilt ja nur zu überprüfen, ob man ein Polygon mit einer Menge von Dominosteinen bedecken kann. Wenn dafür eine Lösung existiert, also wenn man eine gegebene Fläche mit gegebenen Dominosteinen komplett auslegen kann, dann wird diese Lösung von einer nicht-deterministischen Turingmaschine in polynomieller Zeit gefunden.

2 - Vorbereitungen

Layout Graph G^L



- man legt das Polygon (‚das Layout‘) auf ein ‚Gitter’
- sämtliche Gitterpunkte innerhalb der überdeckten Polygonfläche, dem Layout, bilden die Knotenmenge V des Layoutgraphen
- die Gitterpunkte werden miteinander verbunden – jeder mit seinem direkten Nachbarn oben/unten bzw. rechts/links
- die auf diese Weise entstandenen Kanten bilden die Kantenmenge E des Layoutgraphen



Perfect Matching M

- ein ‚Perfect Matching‘ M eines Graphen G bedeckt alle Knoten des Graphen, wobei jeder Knoten jeweils nur zu genau einer Kante des Matchings gehört

2 - Vorbereitungen

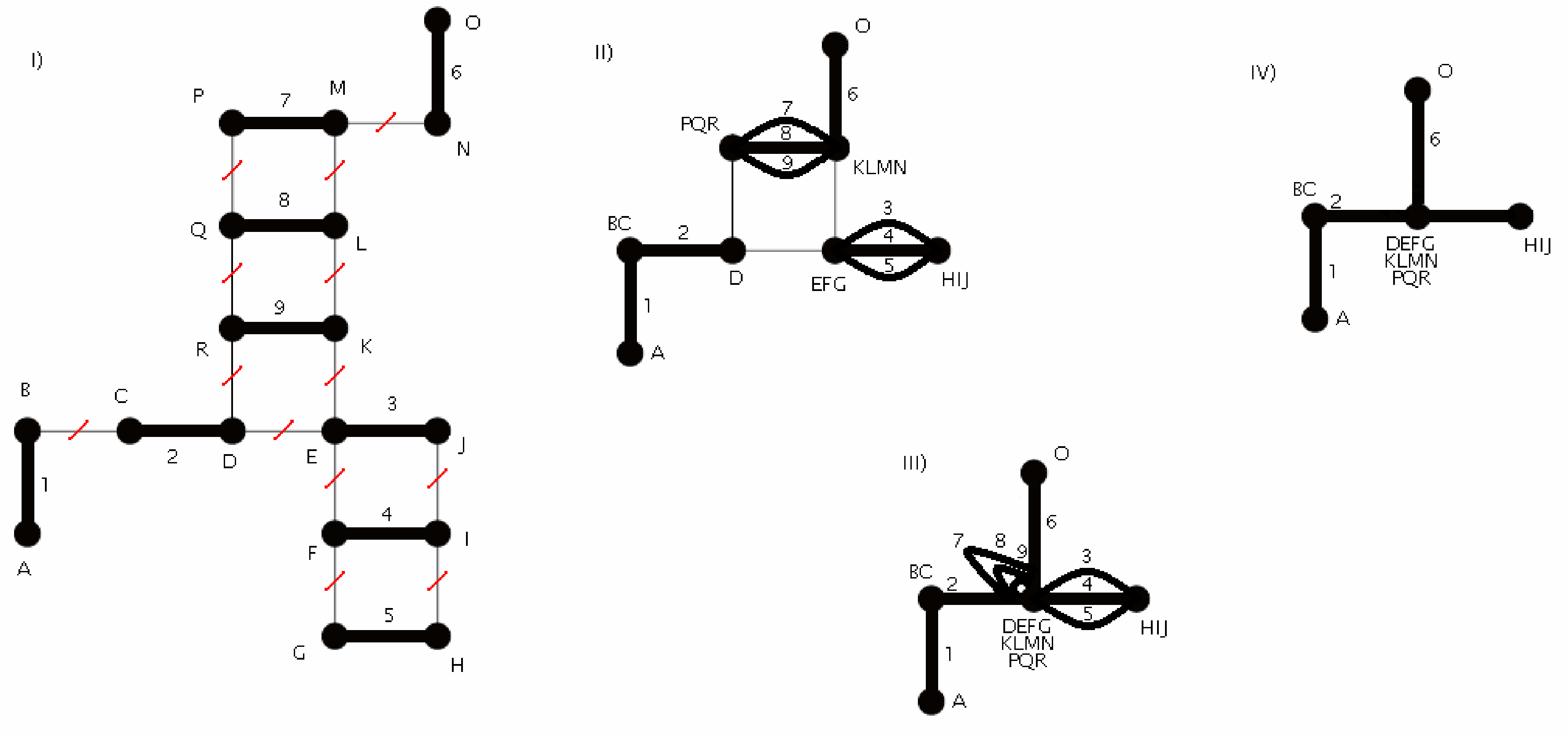
Verkürzter Graph $G^C(M)$

- sei ein Layoutgraph G^L und ein in ihm liegendes ‚Perfect Matching' M gegeben
- der verkürzte Graph G^C (M) ergibt sich aus dem Verkürzen jeder Kante in G^L, die nicht zu M gehört

Verkürzen einer Kante (v, w):

- man entfernt die Kante (v, w)
- wenn $v \neq w$, dann werden v und w ebenfalls entfernt und durch einen neuen Knoten x ersetzt
- x wird nun Nachbar von allen Knoten, die vorher zu v und w benachbart waren

Verkürzter Graph $G^C(M)$



3-Färbbarkeit

Das Problem, zu untersuchen, ob ein Graph 3-färbbar ist - ob man die Knoten eines gegebenen Graphen G so mit 3 Farben einfärben kann, dass zwei benachbarte Knoten nicht dieselbe Farbe haben - ist bekannterweise NP-schwer.

3 – Domino Tiling ist NP-schwer

Die Idee:

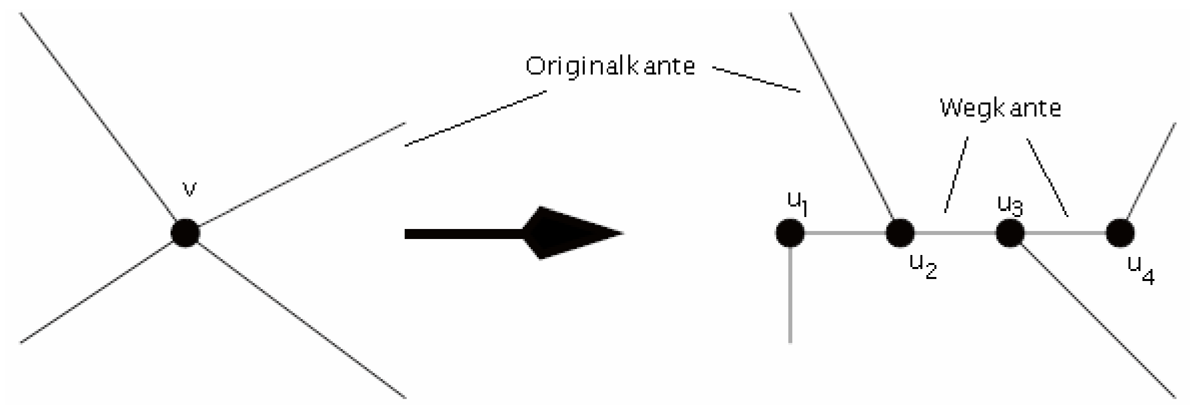
Gegeben sei ein Graph G. Es soll untersucht werden, ob G 3-färbbar ist.

Wenn wir aus G einen Layoutgraphen konstruieren können und G 3-färbbar ist, dann ist auch der Layoutgraph 3-färbbar aus dem wir ein Domino Tiling ableiten können - somit wäre Domino Tiling NP-schwer.

Wir können voraussetzen, dass jeder Knoten von G wenigstens 3 ausgehende Kanten besitzt. Auch dann ist die Frage, ob G 3-färbbar ist, NP-schwer. (Zu finden in weiterführender Literatur)

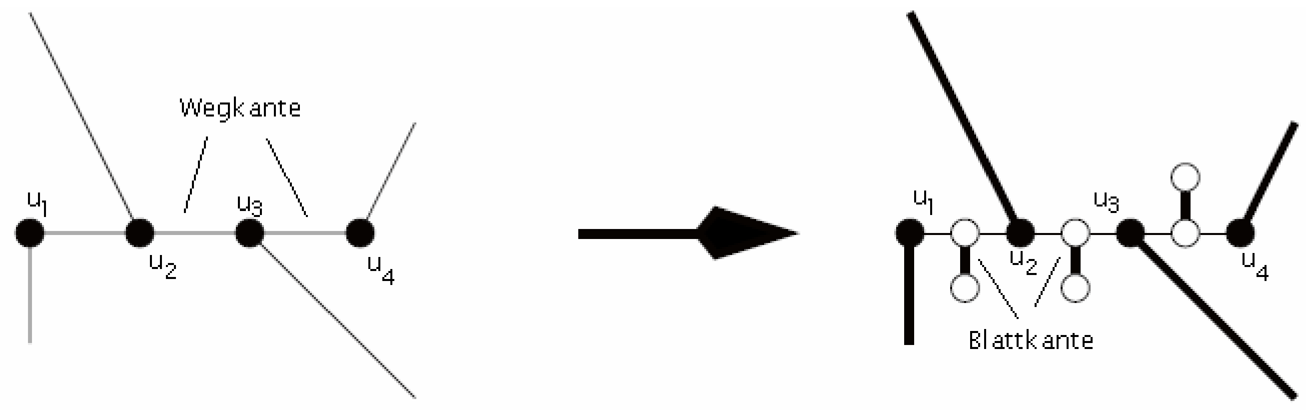
Erzeugen von G^L aus G in zwei Schritten – Schritt 1

- sei G‘ ein Graph, den man durch Modifizieren von G erhält, indem jeder Knoten in einen Weg aufgetrennt wird
- wenn v ein Knoten mit den Nachbarn $w_1, \ldots, w_k$ ist, dann wird v durch einen Weg $u_1, \ldots, u_k$ ersetzt, wobei u_i mit w_i benachbart wird
- die entstandenen Kanten im Weg heißen Wegkanten, die anderen Kanten heißen Originalkanten



Erzeugen von G^L aus G in zwei Schritten – Schritt 2

- sei nun G^L der Graph, den wir aus G‘ erhalten, indem jede Wegkante durch Einfügen eines Knotens geteilt wird
- an diesen eingefügten Knoten wird ein Knoten vom Grad 1 angehangen
- die Kanten zu den Knoten vom Grad 1 heißen Blattkanten



3 – Domino Tiling ist NP-schwer

Behauptung 1:

G^L besitzt ein eindeutiges ‚Perfect Matching‘ bestehend aus den Blattkanten und den Originalkanten.

Behauptung 2:

G^C ist 3-färbbar genau dann, wenn G 3-färbbar ist.

Beweis 1:

G^C erhält man aus G^L, indem alle Kanten verkürzt werden, die nicht zum (eindeutigen) ‚Perfect Matching‘ gehören. Im Einzelnen werden alle Wegkanten verkürzt – das Aufteilen der Knoten wird rückgängig gemacht. Demzufolge entspricht G^C fast G, außer dass von den Knoten Blattkanten ausgehen, an denen Knoten vom Grad 1 hängen.

3 – Domino Tiling ist NP-schwer

Beweis 2:

Wenn G^C 3-färbbar ist, dann ist es auch G, da G ein erzeugter Untergraph ist.

Umgekehrt, sei G 3-färbbar. Dann werden für jeden Knoten v die an v hängenden Knoten vom Grad 1 mit einer Farbe gefärbt, die nicht v hat. Somit ist auch G^C 3-färbbar.

Diese beiden Behauptungen werden gebraucht für …

3 – Domino Tiling ist NP-schwer

Satz 1:

Das Überprüfen, ob ein Graph mit einer bestimmten Menge von Dominos zerlegt werden kann, ist NP-schwer, selbst wenn für die Dominos nur 3 Farben verwendet werden.

Beweis:

Sei ein Graph G für den die Existenz der 3-Färbbarkeit geprüft werden soll gegeben. Aus diesem wird wie zuvor beschrieben ein Layout-Graph G^L konstruiert.

Sei D eine Menge von Dominosteinen mit 3 Farben c_1, c_2, c_3 und für $c_i \neq c_j$ fügen wir „ausreichend viele“ Dominosteine hinzu.

(Ausreichend viele bedeutet hier so viele, dass man nicht in die Situation kommen, keine mehr zu haben. Wenn der Layoutgraph 2n Knoten hat, dann benötigt man insgesamt n Dominos. „Ausreichend viele“ bedeutet also von jedem möglichen Domino n viele.)

3 – Domino Tiling ist NP-schwer

Angenommen, G^L besitzt ein Domino Tiling mit einer Färbefunktion c.

Jeden Knoten in G^C erhält man durch Verkürzen von Knoten w_1, …, w_k, aus G^L, welche durch Kanten verbunden sind, die nicht zum ‚Perfect Matching' M gehören. Daher müssen die Knoten w_1, …, w_k alle dieselbe Farbe haben: Setze $c(v) = c(w_1)$.

Jede Kante in G^C entspricht einem Domino im Domino Tiling. Da es keine einfarbigen Dominosteine geben soll, ist diese Färbung von G^C eine 3-Färbung von G^C.

Nach Behauptung 2 ist somit auch G 3-färbbar.

Die Gegenrichtung ist ähnlich. Wenn G 3-färbbar ist, dann ist es auch G^C. Nun wird jedem Knoten in G^L die Farbe zugewiesen, die der Knoten in G^C besitzt, zu dem er verkürzt wurde. Alle Kanten, die zum Matching gehören, haben nun verschieden gefärbte Endpunkte und entsprechen einem Domino. Das führt zu einem gültigen Domino Tiling, da jeder Typ ausreichend oft vorhanden ist.

3 – Domino Tiling ist NP-schwer

Bisher ist man von einem beliebigen Graphen ausgegangen. Die ursprüngliche Frage geht jedoch von einem gegebenen Layout aus.

Beschäftigen wir uns nun mit Graphen, die aus einem Layout hervorgehen.

Hierfür bedarf es wieder einiger Vorbereitungen.

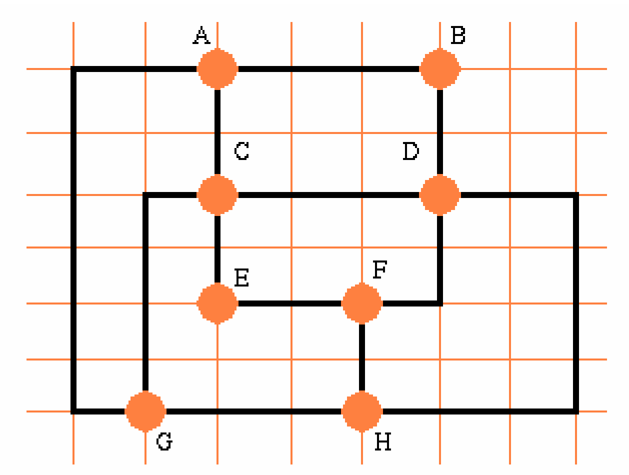
Planarer Graph mit Maximalgrad 4

Ein planarer Graph ist ein Graph, der sich in der Ebene darstellen lässt, ohne dass sich die Kanten schneiden. Dass 3-Färbbarkeit bei Graphen mit Maximalgrad 4 weiterhin NP-schwer ist, wird in einem kommenden Vortrag gezeigt.

Planares orthogonales Darstellen

Ein Graph G wird so gezeichnet, dass sich die Kanten nicht überschneiden und jede Kante besteht aus einer Folge von horizontalen und vertikalen Linien.

Die komplette Länge aller Kanten ist polynomiell.



3 – Domino Tiling ist NP-schwer

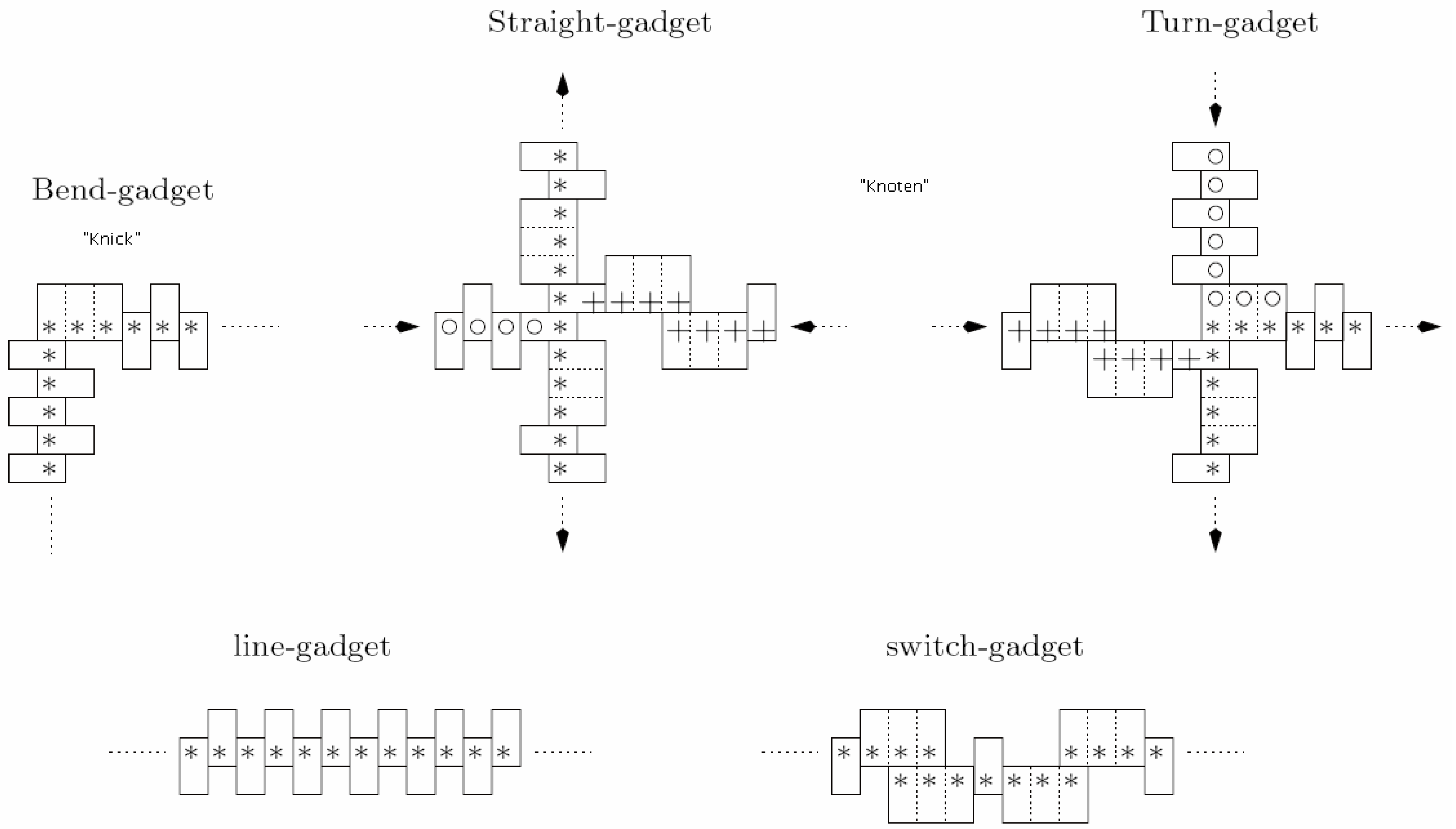
Sei G ein Graph, den wir 3-färben wollten. Sei G ein planarer Graph und abgebildet ist er mittels des planaren orthogonalen Darstellens.

Das Layout erhalten wir durch Erweitern dieser Darstellung von G.

Zunächst strecken wir die Darstellung dergestalt, dass jede einzelne Zeile und Spalte durch viele Zeilen und Spalten (so ~20) ausgetauscht wird, um viel Platz zwischen parallelen Kanten zu erhalten.

Anschließend wird jede Linie, jeder Knoten und jede Ecke durch eine der folgenden Vorrichtungen ersetzt:

3 – Domino Tiling ist NP-schwer



3 – Domino Tiling ist NP-schwer

Diese Vorrichtungen (‚Gadgets‘) haben spezielle Eigenschaften.

Behauptung 3:

Vorausgesetzt, es gibt keine einfarbigen Dominosteine. Dann besitzt jede Vorrichtung ein eindeutiges ‚Perfect Matching‘ entsprechend der Anordnung der Vorrichtung.

Beweis:

Für jeden Knoten vom Grad 1, beinhaltet jedes ‚Perfect Matching‘ die ausgehende Kante. Unter Berücksichtigung dieser Kanten, behalten wir in jeder Vorrichtung ein paar ‚2x3 Rechtecke‘, welche möglicherweise miteinander verbunden sind. Diese Untergraphen haben mehr als ein ‚Perfect Matching‘, aber bei allen Legemöglichkeiten außer einer würde man einfarbige Dominosteine verwenden müssen.

3 – Domino Tiling ist NP-schwer

Behauptung 4:

Angenommen, es gibt ein Domino Tiling der Vorrichtungen ohne einfarbige Dominosteine. Dann müssen alle Felder mit demselben Zeichen auch dieselbe Farbe haben. Des Weiteren haben Felder mit dem Zeichen ‚*' eine andere Farbe als Felder mit dem Symbol ‚+' oder ‘o'. Felder mit ‚+' und ‚o' können, müssen aber nicht dieselbe Farbe haben.

Beweis:

Nach Behauptung 3 gibt es nur ein mögliches ‚Perfect Matching' M. Daher sind Stellen mit demselben Zeichen untereinander mit Kanten verbunden, die nicht in M liegen und sie müssen dieselbe Farbe haben.

Da es zwar Dominos (*, +) und (*, o) gibt aber keinen Domino (+, o), ergibt sich die Forderung nach Verschiedenheit sofort.

3 – Domino Tiling ist NP-schwer

Man unterscheidet beim Domino Tiling abhängig von der Anzahl der verwendeten Dominosteine zwei Probleme:

Partielles Domino Tiling

Beim partiellen Domino Tiling können wir von beliebig vielen Dominos ausgehen.

Exaktes Domino Tiling

Beim exakten Domino Tiling muss jeder Dominostein genau einmal in der Anordnung verwendet werden. Demzufolge muss die Anzahl der Dominos genau der Hälfte der Knoten des Layoutgraphen entsprechen.

3 – Domino Tiling ist NP-schwer

Satz 2:

Partielles Domino Tiling ist NP-schwer, selbst wenn nur 3 Farben verwendet werden.

Beweis:

Sei G ein planarer Graph mit Maximalgrad 4, für welchen wir eine 3-Färbung finden wollen. Wir erzeugen uns eine planare orthogonale Darstellung und strecken sie wie zuvor beschrieben.

Nun wird aus G ein gerichteter Graph gemacht, so dass jeder Knoten höchstens 2 ausgehende und 2 eingehende Kanten besitzt.

(Dies kann man erreichen, indem man durch G durch Hinzufügen von Kanten mittels Eulerschem Kreis durchgeht, den Kanten dabei eine Richtung gibt und abschließend die hinzugefügten Kanten wieder entfernt.)

3 – Domino Tiling ist NP-schwer

Nun wird die Darstellung durch unsere Vorrichtungen ersetzt.

Jede Ecke wird durch ein ‚Bend'-Gadget ersetzt, gegebenenfalls so gedreht, dass die Linien mit ‚*' mit den angrenzenden Kantensegmenten zusammen passen.

Jeder Knoten v wird durch ein ‚Straight'-Gadget oder ‚Turn'-Gadget ersetzt, abhängig von der Ausrichtung der Knoten bzw. der an v anliegenden Kanten.

Wenn v zwei ausgehende Kanten besitzt und beide dieselbe Richtung haben, dann wird das ‚Straight'-Gadget verwendet, sonst das ‚Turn'-Gadget.

Auch hier müssen die Vorrichtungen gegebenenfalls gedreht werden.

Bei Knoten mit weniger Kanten lässt man entsprechende Aus-/Eingänge weg.

Schließlich wird die Darstellung durch die ‚Line'-Gadgets komplettiert, welche die Kanten darstellen. Durch Kombination der ‚Line'-Gadgets mit ‚Bend'-Gadgets oder ‚Vertex'-Gadgets können 2 x 2 Karrees entstehen, welche jedoch die Eindeutigkeit des Perfect Matching zerstören würden. Daher werden Teile des ‚Line'-Gadgets' mit dem ‚Switch'-Gadget dargestellt, welches mit einem nach unten stehenden Domino endet.

3 – Domino Tiling ist NP-schwer

Wir haben nun eine vollständige Beschreibung vom Layout L, die Dominos D haben drei Farben c_1, c_2 und c_3 und für jedes $c_i \neq c_j$ haben wir ausreichend viele Dominos (c_i, c_j).

Es gibt keine einfarbigen Dominos, also gelten die Behauptungen 3 und 4 und wir haben ein Perfect Matching M sowie einen verkürzten Graphen G^C.

Angenommen, wir haben ein farbiges Domino Tiling mit diesen Dominosteinen. Sei für jeden Knoten v c(v) die Farbe, welche für die Stellen, die mit ‚*' markiert sind. Dies führt zu einer 3-Färbung von G.

Wenn (v, w) eine Kante ist, dann wurde sie anfangs auf eine Art gerichtet, sagen wir v -> w. C(v) und c(w) sind somit die Farben für ‚*' in den Vorrichtungen für die Knoten v und w. Aber: c(v) wird in der Darstellung entlang der Kante (v, w) durch die ‚Line'-, ‚Bend'- und ‚Switch'-Gadgets fortgesetzt. Am Ende erreicht es die Vorrichtung von w als ‚+' oder ‚o'.

Daher gilt $c(v) \neq c(w)$ nach Behauptung 4.

3 – Domino Tiling ist NP-schwer

In der umgekehrten Richtung: Sei eine 3-Färbung des Graphen G gegeben, dann erhalten wir ein Domino Tiling, indem wir ‚*' bei der Vorrichtung von v die Farbe von v geben. Das benötigt keine einfarbigen Dominos und da wir ausreichend viele Dominos von allen Farben haben, führt dies zu einer Domino Zerlegung.

Zur Erinnerung: die Darstellung von G besitzt eine polynomielle komplette Kantenlänge. Somit ist die Konstruktion von G polynomiell groß.

Dies vervollständigt den Beweis der NP-Schwere.

Korollar 1:

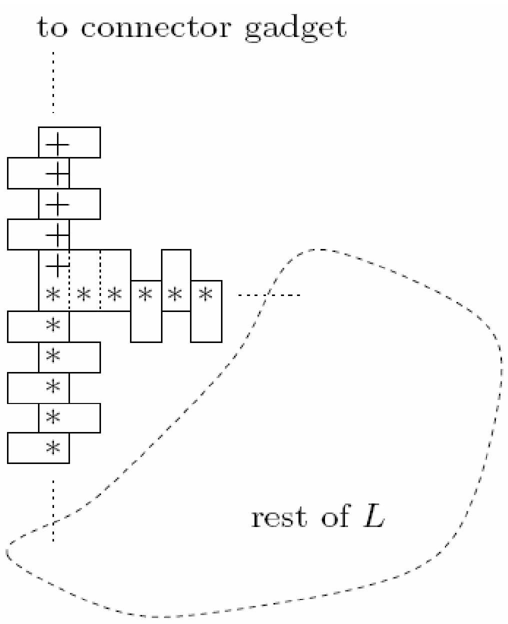
Exaktes Domino Tiling ist NP-schwer, selbst wenn nur 3 Farben verwendet werden.

Beweis:

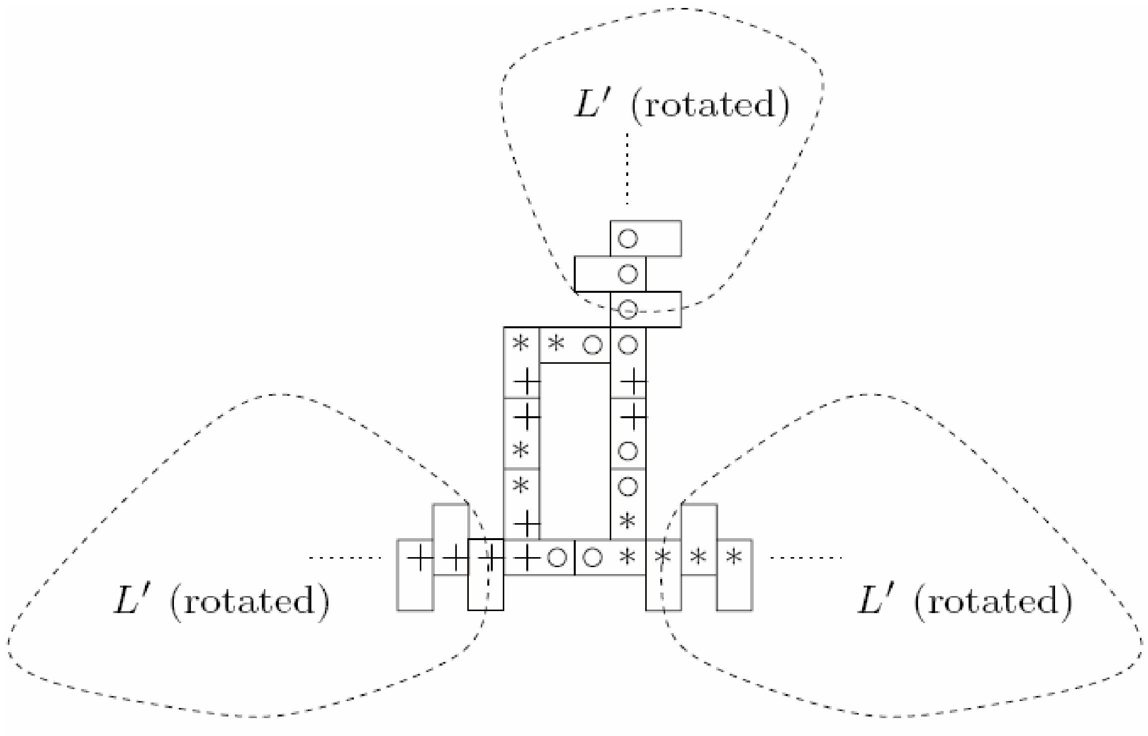
Sei L das Layout vom vorherigen Beweis.

An dieses wird ausreichend oft ein ‚Line'-Gadget angehangen (z.B. an die Stelle des 2 x 3 Rechteckes in einem ‚Bend'-Gadget) ohne dass es Einfluss darauf hat, ob L ein Domino Tiling besitzt oder nicht.

Das neue Layout wird L' genannt und wir setzen voraus, dass es 2n' Knoten besitzt.



Sei nun L‘‘ das Layout, welches aus dem 3maligen Kopieren von L‘ entsteht, wobei die Kopien von L‘ gedreht und mittels einer Vorrichtung bestehend aus 9 Dominos miteinander verbunden sind.



3 – Domino Tiling ist NP-schwer

Die Menge der Dominos ist dieselbe wir beim Beweis zuvor, mit der Ausnahme, dass jeder Dominotyp (n‘ + 3)-mal existiert.

Wir besitzen also 3n‘ + 9 Dominosteine insgesamt und L‘‘ besitzt 6n‘ + 18 Knoten. Falls also ein Domino Tiling existiert, dann muss es exakt sein.

Wenn L‘‘ ein Domino Tiling besitzt, dann auch L‘ und somit L.

Umgekehrt: Wenn L ein Domino Tiling besitzt, dann besitzt L‘ ein Domino Tiling, indem die angehangenen ‚Line‘-Gadgets beliebig gefärbt werden.

Wir haben dafür ausreichend viele Dominos, da L‘ n‘ Dominos benötigt.

Nun können wir also die 3 Kopien von L‘ mit Dominos zerlegen, wobei die beiden anderen Kopien jeweils um eins versetzt eingefärbt werden. Das Zerlegen der 3 Kopien verwendet jeweils exakt n‘ Dominos von jeder Art und die 3 Enden der ‚Line‘-Gadgets der Kopien der L‘ sind jeweils mit einer anderen Farbe eingefärbt.

3 – Domino Tiling ist NP-schwer

Letztendlich kann die Verbindungs-Vorrichtung wie in der Abbildung zerlegt werden. An den 3 Verbindungs-Punkten zu den Kopien der L‘ haben wir 3 verschiedene Farben und daher ein Domino Tiling von L‘‘.

4 – Zusammenfassung

Es wurde gezeigt, dass das Zerlegen eines Polygons mit Dominos NP-vollständig ist.

Zunächst sind wir von einem beliebigen Graphen ausgegangen, den wir in einen Layoutgraphen überführt haben. Durch die Existenz eines ‚Perfect Matchings' wurde dieser Layoutgraph verkürzt und für diesen verkürzten Graphen haben wir gezeigt, dass er 3-färbbar ist.

Abschließend haben wir unter Zuhilfenahme von planaren Graphen und planaren orthogonalen Darstellungen eines Graphen gezeigt, dass wir beliebige Layouts darstellen können. Durch Abbilden auf 3-Färbbarkeit haben wir wie zuvor gezeigt, dass sowohl partielles Domino Tiling, als auch exaktes Domino Tiling unter Verwendung von nur 3 Farben NP-schwer ist.

5 – Literaturangaben und Quellenangaben

Studie: „The Complexity of Domino Tiling“
Autor: Therese Biedl

Bildmaterial und weitere Literatur:

- http://de.wikipedia.org/wiki/Domino_%28Spiel%29
- http://wwwipr.ira.uka.de/~lehre/EInfo/Vorlesungsfolien/Info1/VL8_2.pdf
- http://www.thi.informatik.uni-frankfurt.de/~klauck/VorlBB05/VorlesungBB05_4.pdf
- http://www.cs.brown.edu/~jjl/BendMinIntro.html