

Das 15er-Puzzle ist NP-Vollständig

Ausarbeitung zum Seminar über Algorithmen und Komplexität

von David Iwanowitsch

Quelle: D. Ratner, M. Warmuth, *The (n^2-1) -Puzzle and Related Relocation Problems*, *J. Symbolic Computation* 10 (1990), p111-137.

Das 15er-Puzzle ist im deutschen auch bekannt als „Schiebepuzzle“. Diese Ausarbeitung beschäftigt sich mit der allgemeineren Form, dem $(n \times n)$ -Puzzle, welche die ursprüngliche Version des Spiels von einem Feld mit der Kantenlänge vier auf ein Feld mit der Kantenlänge n erweitert, es wird daher auch (n^2-1) -Puzzle genannt. Es wird gezeigt, dass das Finden einer Zugfolge, die mit möglichst wenig Schritten zur Lösung führt, ein NP-Vollständiges Problem ist.

Einführende Definitionen

Das Spiel: n^2 eindeutige Felder, eine leere Kachel, n^2-1 Kacheln nummeriert von 1 bis n^2-1 . Jedes Feld hat genau eine Kachel.

Ein Zug ist der Austausch der leeren Kachel mit einer horizontal oder vertikal angrenzenden anderen Kachel.

Ziel: Transformieren einer Konfiguration in eine andere.

Erreichbarkeit:

Eine Konfigurationen ist nicht von jeder anderen aus erreichbar.

Für jede Kachel t sei $\pi(t)$ die Kachel, die nach der Transformation an der Stelle von t liegt. Jede Vertauschung kann man in Kreise der Form (t_1, t_2, \dots, t_q) aufspalten wobei $t_i = \pi(t_{(i+1) \bmod q})$ ist. Ein solcher Kreis ist gerade, wenn seine Größe gerade ist.

Satz: eine Konfiguration ist von einer anderen Konfiguration auf der die leere Kachel die gleiche Position hat *erreichbar*, wenn ihre Vertauschung aus einer geraden Anzahl von geraden Kreisen besteht. (Lässt sich in $O(n^2)$ prüfen.)

Daraus ergibt sich auch, dass wenn man in zwei zueinander erreichbaren Konfigurationen zwei nicht leere Kacheln vertauscht, so sind sie nicht mehr erreichbar.

Algorithmus: Es gibt einen Algorithmus, der zwei erreichbare Konfigurationen in $O(n^3)$ Schritten ineinander umformt. (Im Beweis wird er als Funktion $p(x)$ verwendet.)

(Nicht) benötigte Züge:

Der Manhattan-Abstand der beiden Orte jeder Kachel von Start- und Zielkonfiguration gibt insgesamt die minimal nötige Anzahl von Zügen an. Wenn nach einem Zug der Abstand kleiner wird, handelt es sich um einen benötigten Zug. Der Gesamtbestand zwischen zwei Konfigurationen A und B wird mit $d(A,B)$ bezeichnet.

Das Problem nPUZ

Für das folgende Entscheidungsproblem wird gezeigt, dass es NP-Vollständig ist, das Problem wird als **nPUZ** bezeichnet:

INSTANZ: Zwei Konfigurationen für ein $n \times n$ Feld, und eine obere Schranke k .

FRAGE: Gibt es einen Weg, die eine Konfiguration in maximal k Schritten in die zweite Konfiguration umzuwandeln?

Da die Regeln von nPUZ sehr einschränkend wirken, wird der Beweis sehr kompliziert. Daher wird zunächst das Problem vereinfacht und beim eigentlichen Beweis auf den leichteren zurückgegriffen. Das vereinfachte Problem wird als **REL** bezeichnet und basiert auf planaren Graphen.

INSTANZ: Ein planarer, gerichteter Graph $G(V,E)$. Eine Menge X von Elementen und eine Ausgangs- und Endkonfiguration (Eine Konfiguration spezifiziert die Lage von jedem Element aus X auf den Knoten aus V). Jedes $e \in E$ kann 0 oder 2 Elemente transportieren (Kapazität).

FRAGE: Gibt es eine Prozedur, die jedes Element aus X aus der Ausgangs- in die Endkonfiguration bringt, wobei jede Kante genau einmal benutzt wird (Eulerscher Pfad) und die Kapazität der Kanten berücksichtigt wird?

REL ist NP-vollständig

Um zu zeigen, dass REL NP-Vollständig ist, muss es auf ein bekanntes Problem aus NP reduziert werden. Dazu wird das Problem 2/2/4-SAT verwendet (NP-Vollständigkeitsbeweis dazu im Anhang der Quelle).

2/2/4-SAT:

INSTANZ: Menge U mit m Variablen und eine Menge C von m Klauseln über U , so dass jede Klausel $c \in C$ genau 4 Literale hat und jedes $u \in U$ in allen Klauseln von C genau viermal vorkommt zweimal normal und zweimal negiert.

FRAGE: Gibt es eine Wahrheitsbelegung für U , so dass jede Klausel in C genau 2 wahre Literale hat?

Konstruktion

Sei $U = \{u_1 \dots u_m\}$ Menge von Variablen, $C = \{c_1 \dots c_m\}$ Menge von Klauseln, U und C definieren eine willkürliche Instanz von 2/2/4-SAT.

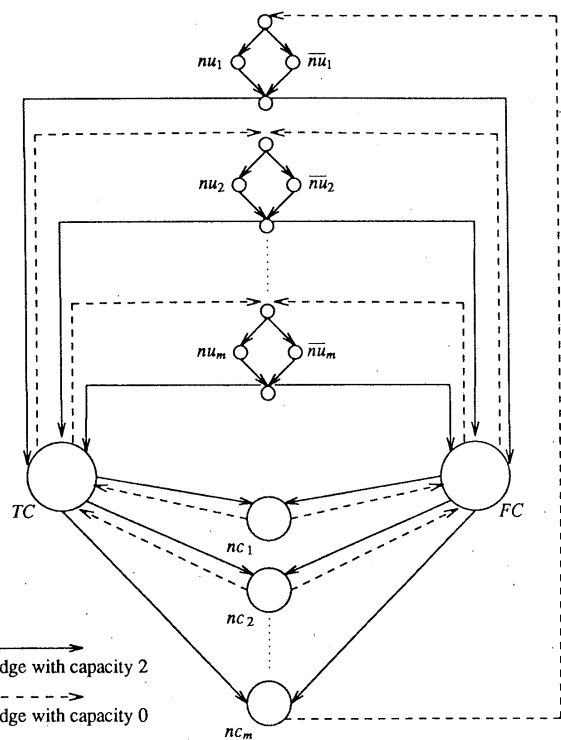
Aus dieser Instanz wird eine Instanz für REL konstruiert, die aus dem Graph $G(V,E)$ mit Kapazitäten 0 und 2 besteht, sowie aus einer Menge X von Elementen, und den beiden Konfigurationen B_1 und B_2 .

Beschreibung des Graphen

Der Graph besteht aus $5m+2$ Knoten und $12m-3$ Kanten. Es gibt vier Gruppen von Knoten:

1. m Rhomben mit je 4 Knoten und 4 Kanten mit Kapazität 2. Der i -te Rhombus entspricht Variable u_i , er enthält die Knoten top_i , nu_i , bot_i , \bar{nu}_i
2. Der Knoten TC „truth collection“
3. Der Knoten FC „false collection“
4. m „Klauselknoten“, wobei der i -te Knoten der Gruppe (nc_i) der i -ten Klausel der booleschen Formel entspricht.

Die Kapazitäten und Kantenverläufe lassen sich aus der Abbildung ablesen.



Beschreibung der Menge X und der Konfigurationen

X besteht aus $8m$ Elementen (jede der m Variablen zweimal negiert und zweimal unnegiert, sowie jeweils 4 Füllelemente), $n_{i,1}$ und $n_{i,2}$ entsprechen den Vorkommen von u_i . $n_{i,3}$ und $n_{i,4}$ denen von \bar{u}_i . $n_{i,5}$ bis $n_{i,8}$ sind Füllelemente, die erst bei der nPUZ Reduktion eine Rolle spielen.

In B_1 sind die eigentlichen Elemente in den Rhomben angeordnet. In B_2 befinden sie sich dann in den Knoten, die zu ihrer Klausel gehören (nc_i).

Beweis: 2/2/4-SAT -> REL

Zunächst werden mit einer Shift-Prozedur alle $n_{i,j}$ Elemente, die die wahren Literale verkörpern aus ihren Knoten in B_1 nach TC verschoben. TC enthält dann $2m$ Literale (aus jedem Rhombus genau zwei Elemente). Aus top_i werden dabei jeweils zwei Füllelemente nach unten verschoben.

Die $2m$ Literale in TC werden jetzt mit einer weiteren Shift-Prozedur, in den für sie in B_2 vorgesehenen Knoten nc_i verschoben.

Als Ergebnis hat jetzt jeder Klauselknoten die beiden Elemente, die den wahren Literalen in B_2 entsprechen.

Von nc_m wird wieder nach oben zu top_1 gegangen und die beiden Shift-Prozeduren erneut angewendet, diesmal werden aber die Elemente, die den falschen Literalen entsprechen nach FC verschoben und die $2m$ Elemente aus FC werden

ebenfalls in die Klauselknoten verschoben.

Jetzt entspricht die Anordnung der Knoten, der in B_2 , und da jede Kante genau einmal mit ihrer vollen Kapazität benutzt wurde, ist ein Eulerscher Pfad abgelaufen worden.

Beweis: REL -> 2/2/4-SAT

Es gibt eine Prozedur, die den Graphen von REL von der Konfiguration B_1 nach B_2 überführt. Dabei müssen die 4 Elemente aus jedem Rhombus nach unten in die Klauselknoten wandern. Dazu müssen hinter bot_i zwangsweise 2 nach TC und 2 nach FC wandern. Da es sonst keinen Eulerschen Pfad geben kann.

Die Wahrheitsbelegung wird wie folgt definiert:

- $f(u_i) = T$, wenn die Prozedur $n_{i,1}$ und $n_{i,2}$ über die Kante (bot_i, TC) transportiert.
- $f(u_i) = F$, wenn die Prozedur $n_{i,1}$ und $n_{i,2}$ über die Kante (bot_i, FC) transportiert.

Die Belegung genügt 2/2/4-SAT: Aufgrund der Kanten und Kapazitäten müssen in jeder Klausel zwei Elemente aus TC und zwei aus FC vorhanden sein, also hat jede Klausel genau zwei wahre und zwei falsche Literale.

Damit ist der Beweis abgeschlossen.

nPUZ ist NP-vollständig

Konstruktion einer nPUZ Instanz

Die konstruierte Instanz ist ähnlich der von REL im vorherigen Abschnitt. Der bei REL gezeigte Graph wird auf das Spielfeld abgebildet.

Eine nPUZ Instanz besteht aus den Konfigurationen B_1 und B_2 für ein $(n \times n)$ -Feld und der oberen Schranke k der Anzahl der Schritte von B_1 nach B_2 .

Jeder Knoten bei REL entspricht einem quadratischen Bereich von Feldern auf dem Brett.

Kanten entsprechen kurzen Sequenzen von dünnen horizontalen und vertikalen Streifen von Feldern, die die Knoten verbinden. (Vergleiche die Abbildungen der Graphen von REL und nPUZ.)

Die durchgezogenen Linien gehören zu Streifen der Höhe drei und haben die Kapazität zwei.

Die gestrichelten Linien haben eine Höhe von einem Feld und Kapazität null.

Alle Knoten (Quadrate) haben eine Seitenlänge von drei Feldern, nur FC und TC haben eine Seitenlänge von 4m (sie werden „große“ Knoten genannt, der Rest „kleine“ Knoten).

Jede Kante ist mindestens 5l lang, l ist abhängig vom Wert u, von dem gezeigt werden wird, dass es eine obere Schranke an nicht benötigten Zügen ist:

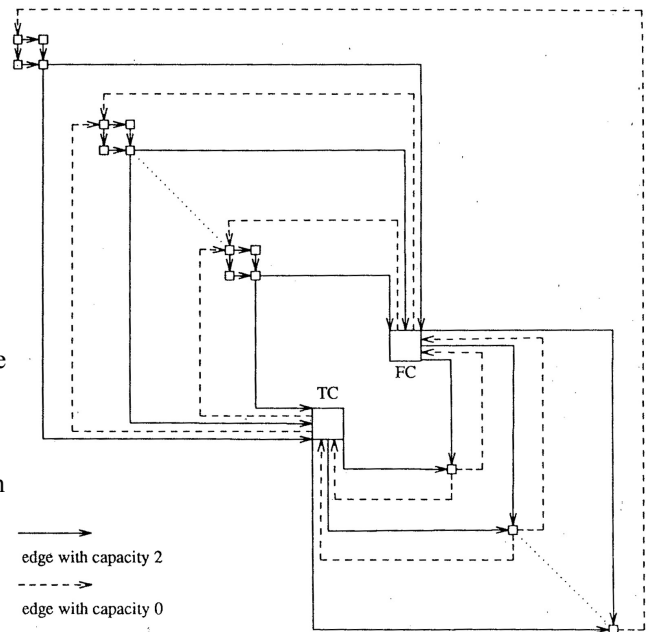
$l = u + 16m + 1$ und $u = 508m + 29m p(6) + 9m p(4m + 3)$ p ist die Funktion, aus dem im ersten Abschnitt genannten Algorithmus.

B_1 und B_2 haben nur in den abgebildeten Knoten und Kanten unterschiedliche Belegungen, also ist jede Kachel außerhalb der Knoten und Streifen in B_1 und B_2 am gleichen Ort.

Man kann den Kanten und Knoten leicht Orte zuweisen, so das die obigen Bedingungen zutreffen. Das kann so erledigt werden, dass die längste Kante und der Abstand zwischen zwei beliebigen Knoten mindestens 5l ist. Und die Seitenlänge des Spielfeldes in $O(ml)$ liegt.

Kodierung der 2/2/4-SAT Instanz in die Feldkonfigurationen B_1 und B_2

Es gibt wieder eine Menge U von m Variablen und eine Menge C mit m Klauseln, die eine willkürliche 2/2/4-SAT Instanz definieren.

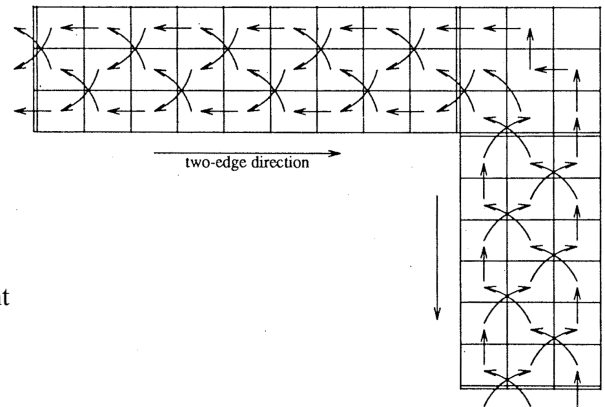


Wie bei REL entsprechen die beiden Vorkommen von u_i den Elementen (Kacheln) $n_{i,1}$ und $n_{i,2}$ angeordnet im Bereich von nu_i in B_1 . Analog dazu \overline{nu}_i mit den Kacheln $n_{i,3}$ und $n_{i,4}$. Ebenfalls wie bei REL werden die $4m$ Elementkacheln verteilt auf die Knoten nc_i in B_2 . Der Bereich von nc_i bekommt die vier Literalkacheln die den Literalen in c_i entsprechen. Die vier Füllelemente $n_{i,5}$ bis $n_{i,8}$ sind in B_1 wieder in top_i angeordnet. In B_2 befinden sie sich dann in nu_i und \overline{nu}_i .

Die relative Anordnung der Kacheln in den Kanten in B_1 und B_2 ist wie folgt:

Bei Kanten mit Kapazität null wird jede Kachel einfach ein Feld nach hinten bewegt (Richtung der Kante beachten). Dabei verlässt eine Kachel die Kante am Ende und eine neue Kachel kommt am Kopfende dazu. dazu wird eine Kachel gewählt die am Kopfende im Knoten anliegt und keine Elementkachel ist.

Die Neuordnung bei Kanten mit Kapazität zwei verzwickter, am besten erklärt es die nebenstehende Abbildung.. Hierbei kommen drei Kacheln hinzu, und drei rutschen aus der Kante hinaus.



Da die Größe der Knoten vernachlässigbar klein ist (durch unsere Wahl von l , wird später gezeigt), ist die exakte Position der Kacheln in ihnen unwichtig und kann leicht korrigiert werden.

Letzten Endes könnte es möglich sein, dass B_2 nicht von B_1 erreicht werden kann, wenn dem so ist, werden einfach zwei beliebige Kacheln in nc_m vertauscht (siehe Erreichbarkeit).

Zum Abschluss der Beschreibung der Instanz wird k auf $d(B_1, B_2) + u$ gesetzt.

Man kann leicht sehen, dass die beiden Konfigurationen und k in polynomieller Zeit von m gebildet werden können. Im Folgenden wird gezeigt, dass 2/2/4-SAT genau dann eine Lösung hat, wenn B_1 in maximal k Schritten zu B_2 transformiert werden kann.

Beweis 2/2/4-SAT -> nPUZ

Die Konfiguration wird durch modifizierte Versionen der bei REL angewendeten Shift-Prozeduren umgewandelt. Der Hauptunterschied liegt darin, dass die Bewegung entlang einer Kante komplexer ist.

Bei Kanten mit Kapazität null („Nuller-Kanten“) liegt die leere Kachel im Knoten am Ende der Kante. und wird schrittweise durch die Kante geschoben, wobei jede Kachel in der Kante um 1 nach hinten rutscht.

Um Kanten mit der Kapazität zwei („Zweier-Kanten“) zu durchlaufen, werden zunächst im Knoten die Kacheln x und y in einer Kette hinter der leeren Kachel angeordnet, die direkt am Übergang zur Kante anliegt. Danach wird die leere Kachel abwechselnd mit der Sequenz S_u und S_d bewegt, $S_u = \langle \text{links, links, hoch, rechts, rechts, rechts, runter} \rangle$ und S_d entspricht S_u mit vertauschtem „hoch“ und „runter“. Für eine r lange Kante müssen S_d und S_u $r+3$ mal angewendet werden, um die Kette $[x, y, \text{leer}]$ in den nächsten Knoten zu verschieben. Diese Bewegung entspricht der in der Abbildung im vorherigen Abschnitt beschriebenen. Also bewegen sich die Kacheln korrekt.

Zweier-Kanten können aber auch Ecken haben, dann gibt es die Drehreihenfolge $\langle \text{links, links, runter, rechts, rechts, hoch, links, runter, runter, rechts} \rangle$ für eine 90° Drehung im Uhrzeigersinn, bzw. mit vertauschtem „hoch“ und „runter“ gegen den Uhrzeigersinn, danach müssen S_u und S_d angepasst (rotiert) werden.

Folgende Details müssen beachtet werden: Nach der Traversierung mit Kapazität c ($0/2$) werden $c+1$ Kacheln über die Knotengrenze in die Kante verschoben. Am Ende der Traversierung werden $c+1$ Kacheln aus dem Knoten in die Kante „gezogen“, wenn diese $c+1$ Kacheln nicht der Anordnung in B_2 entsprechen, wird der Knoten temporär um sie erweitert, und die Kacheln werden gemäß B_2 angeordnet.

Wenn die leere Kachel einen Knoten nach den Shift-Prozeduren das letzte Mal verlässt, muss dieser der Anordnung in B_2 entsprechen. Dank der Füllelemente wird jeder Knoten, außer die Klauselknoten über eine Zweier-Kante verlassen, so dass eine temporäre Erweiterung problemlos möglich ist.

Die letzte Anordnung in einem Klauselknoten nc_j ($0 < j < m$) wird folgendermaßen organisiert: über eine Zweier-Kante gelangt man von FC in den Knoten, und verlässt ihn über eine Nuller-Kante. Wenn am Ende die Anordnung nicht wie in B_2 erfolgen kann, so macht man alles rückgängig bis FC vertauscht in der Kette x und y , und macht damit weiter, jetzt ist die Neuordnung nach B_2 möglich.

Zuletzt muss der Knoten nc_m wie in B_2 angeordnet werden: beim Eintritt in nc_m ist das komplette Brett wie in B_2 angeordnet, nur nc_m noch nicht. Angenommen der Bereich von nc_m kann nicht wie in B_2 angeordnet werden, dann geht es bist auf eine Vertauschung, und es gibt einen Widerspruch, da B_1 und B_2 voneinander erreichbar sind (nach

Konstruktionsvorschrift), also ist das nicht möglich.

Damit sind die Anpassungen an den Shift-Prozeduren vollständig beschrieben. Jetzt muss noch gezeigt werden, dass Insgesamt nur $k = d(B_1, B_2) + u$ Schritte nötig sind.

Alle Schritte auf Nuller-Kanten sind nötig, da sie die Steine auf der Kante von B_1 nach B_2 bringen.

Genauso sind die 7-Schrittsequenzen nötig, da sie die Steine auf den Kanten und x und y ihrem Ziel näher bringen.

Da der letzte Zug auf einer Nuller-Kante Veränderungen im Knoten vornimmt, ist er unter Umständen nicht nötig, es kommt zu maximal $4m-3$ unnötigen Zügen (=Anzahl Nuller-Kanten).

Aus dem gleichen Grund werden evtl. die ersten und letzten 3 (=6) Anwendungen der 7-Schrittsequenzen jeder Zweier-Kante nicht benötigt. Bei $12m$ Zweier-Kanten ergibt das $12 \cdot 6 \cdot 7m$ Züge (=504m).

Insgesamt gibt es an den Kanten also nicht mehr als $508m$ unnötige Schritte.

Als letztes müssen noch die Schritte, die nötig sind um die Knoten und Kanten (mit Erweiterung) umzusortieren, gezählt werden. Es werden $4m$ Knoten über eine Zweier-Kante verlassen und für jede Neusortierung einer 3×3 großen Kante werden $p(3)$ Schritte benötigt. Es gibt $5m$ „kleine“ Knoten mit 3×3 Feldern, die jeweils maximal fünfmal neu sortiert werden müssen (einmal für jede ein- und ausgehende Kante, einmal zum Schluss). Die Knotengröße ist maximal 6×6 daraus ergeben sich $29m$ $p(6)$ Schritte ($29 = 4 + 5^2$).

Die „großen“ Knoten haben die Größe $4m$ und insgesamt $4m-2$ ein- oder ausgehende Kanten. Außerdem muss FC bis zu $m-1$ mal die Kettenelemente vertauschen. Die Kosten pro Neuordnung und Vertauschung betragen $p(4m+3)$.

Gesamtanzahl der Schritte: $d(B_1, B_2) + 508m + 29m p(6) + 9m p(4m+3) = d(B_1, B_2) + u = d(B_1, B_2) + k$.

Damit ist der erste Teil des Beweises erbracht.

Beweis: nPUZ -> 2/2/4-SAT

LSG sei eine Sequenz von maximal k Schritten, die B_1 zu B_2 umformt. Es soll gezeigt werden, dass aus LSG eine erfüllende Belegung für 2/2/4-SAT hervorgeht.

Dazu wird gezeigt, dass die leere Kachel und die Elementkacheln gezwungen sind sich entlang den Kanten zu bewegen (von einer vernachlässigbar kleinen Anzahl Zügen abgesehen) und dass solange k nicht viel größer, als die Anzahl der benötigten Schritte ist, die Anzahl der durch eine Kante wandernden Kacheln ihrer Kapazität entspricht, außerdem wird so verhindert, dass die leere Kachel ein zweites mal durch eine Kante wandert.

Sobald diese Aussagen bewiesen sind, kann man die Spielregeln außer Acht lassen und die Instanz von nPUZ kann wie eine Instanz von REL behandelt werden. Da bereits gezeigt wurde, wie eine Instanz von REL in eine gültige Belegung für 2/2/4-SAT umgewandelt werden kann, ist der Beweis dann gültig.

Da LSG aus $d(B_1, B_2) + u$ Schritten besteht, können höchstens u unnötige Schritte in der Sequenz vorhanden sein. Bei der Brettkonstruktion mit Längeneinheit $l = u + 16m + 1$ kann die leere Kachel, also nur eine begrenzte Anzahl von Feldern auf dem Brett berührt haben.

Definition 1: Die *Umgebung* einer Kante erhält man durch Erweiterung der einzelnen Kantenstreifen um l Felder auf jeder Seite. Dadurch besteht die Umgebung von Kanten mit Kapazität c aus Streifen mit der Breite $2l + c + 1$.

Die *Umgebung* eines „kleinen“ Knotens erhält man, indem man den Knoten auf jeder Seite um l Felder erweitert, damit umfasst die Umgebung $(2l + c)^2$ Felder.

Die *Umgebung* von TC/FC wird in zwei Schritten konstruiert: TC/FC seien Rechtecke die TC/FC und alle ein und ausgehenden Streifen von Kanten enthalten. Die Umgebung besteht dann aus TC/FC in jede Richtung um l Felder erweitert.

Beachte: Alle Umgebungen sind disjunkt, da die Länge der Streifen der Kanten und der Abstand zwischen zwei Knoten mindestens $5l$ beträgt.

Lemma 1: In LSG ist die leere Kachel immer in der Umgebung einer Kante oder eines Knotens.

Beweis: Alle Orte außerhalb der Kanten und Knoten sind in B_1 und B_2 identisch. Jeder Ort L außerhalb einer Umgebung ist mindestens $l+1$ Felder entfernt von jedem Knoten und jeder Kante. Damit die leere Kachel nach L gelangt müssten $l+1$ Kacheln bewegt werden, die in B_1 und B_2 unverändert sind, also wären das $l+1$ unnötige Züge. Das ist ein Widerspruch, zur oberen Schranke k , da $l+1 > u$. \square

Jetzt wird beschrieben, wie das entlang laufen an einer Kante abläuft, dazu wird die Umgebung der Kanten in 3 Segmente zerlegt und gezeigt, dass die leere Kachel nicht rückwärts durch diese Segmente laufen kann:

Definition 2: Sei e eine Kante von ihrem Startknoten n zum Kopfknoten n' . Die Umgebung von e wird in *Eingang*, *Zentrum* und *Ausgang* aufgeteilt. Der Eingang besteht aus allen Knoten die höchstens l Felder von der Umgebung von n

entfernt sind. Der Ausgang analog dazu mit dem Knoten n' . Das Zentrum besteht aus dem Rest zwischen Eingang und Ausgang.

Lemma 2: Wenn in LSG die leere Kachel im Zentrum einer Kante ist dann ist die nächste Knotenumgebung, die sie besucht, die des Kopfknotens. Wenn in LSG die leere Kachel in der Umgebung eines Knotens ist, dann ist das nächste Zentrum einer Kante, das sie passiert, das einer vom Knoten abgehenden Kante.

Beweis: Es genügt zu zeigen, dass die leere Kachel weder rückwärts durch den Eingang noch durch den Ausgang wandern kann. Dieses wird für den Eingang gezeigt, der Beweis für den Ausgang ist identisch. Angenommen die leere Kachel bewegt sich rückwärts durch den Eingang, sei $Q=L_1, \dots, L_q$ die Sequenz der Felder des Eingangs die von der leeren Kachel besucht wurden, als sie aus dem Zentrum zur Umgebung des Endknotens gewandert ist. Sei L_{i_1}, \dots, L_{i_k} die Untersequenz von Q , so dass L_{i_j} das erste Feld ist, das j Felder entfernt von der Grenze zwischen Zentrum und Eingang ist ($1 \leq j \leq k$). Jedes Feld von L_{i_j} ist eindeutig und enthält unterschiedliche Kacheln (t_{ij}). Nach Definition von Q folgt, dass alle Kacheln $T=\{t_{ij}\}$ vorwärts wandern. Wenn t_{ij} keine Elementkachel ist, dann war der erste Zug der leeren Kachel nach L_{i_j} ein unnötiger Zug. Nur Elementkacheln können in einem benötigtem Zug in Richtung Eingang wandern. Es gibt $8m$ dieser Kacheln, also $l-8m$ unnötige Züge wurden ausgeführt während die Kante rückwärts durchwandert wurde. Das ist ein Widerspruch, da $u < l-8m$. \square

Das Durchwandern einer Kante in LSG entspricht also der leeren Kachel, die von der Umgebung des Endknotens bis zur Umgebung des Kopfknotens durch das Zentrum der Kante wandert.

Es muss nur noch gezeigt werden, dass jede Kante genau einmal durchwandert wird und dabei die richtige Anzahl an Kacheln mitgenommen werden.

Lemma 3: Jede Kante wird genau einmal durchlaufen.

Beweis: Durch das vorherige Lemma und da die Kacheln in den Kanten in B_2 anders angeordnet sind, muss jede Kante mindestens einmal durchlaufen werden. Zunächst wird gezeigt, dass Nuller-Kanten genau einmal durchlaufen werden: Es wird von der ersten Schrittsequenz der leeren Kachel durch den Eingang der Kante ausgegangen. Sei L_{i_j} das erste Feld während der Folge, das einen Abstand j von der Grenze der Endknotenumgebung und der Eingangsumgebung hat. T und t_{ij} seien wie im Lemma 2 konstruiert, wenn die L_{i_j} Felder das erste mal besucht werden, werden die l Felder aus T ein Feld rückwärts verschoben. Falls es eine zweite Sequenz gibt, die durch den Eingang wandert, dann wird eine zweite Menge T' mit l Feldern rückwärts verschoben. Die einzigen Kacheln, die in der Eingangsumgebung mit benötigten Zügen rückwärts verschoben werden können sind die l Kacheln, die in B_1 in der Eingangsumgebung liegen, und evtl. Elementkacheln. Die ersteren können nur einmal rückwärts verschoben werden, das führt zu l benötigten Zügen. Jede der $8m$ Elementkacheln kann in beiden Mengen vorhanden sein, das führt zu $2l-l-16m > u$ Kacheln die unnötigerweise bewegt wurden. Das ist ein Widerspruch.

Es muss noch gezeigt werden, dass die Zweier-Kanten genau einmal durchwandert werden. Jeder Rhombus hat zwei eingehende Kanten (außer der erste, da aber hier gestartet wird, kann man sich eine zweite Kante denken). Diese Kanten sind Nuller-Kanten, da die leere Kachel genau einmal durch sie wandert, führt das dazu, dass die leere Kachel alle anliegenden Kanten am Rhombus genau einmal durchläuft. Für jeden Knoten n_i existieren zwei eingehende und zwei ausgehende Kanten, die ausgehenden sind Nuller-Kanten, also können die eingehenden auch nur einmal durchlaufen werden. \square

Lemma 4: Wenn eine Kante mit Kapazität c in LSG durchlaufen wird, dann ist die Anzahl der Elementkacheln die durch das Zentrum laufen, während die leere Kachel die Kante durchläuft gleich der Kapazität c .

Beweis: Sei e eine Kante mit Kapazität c . Führe LSG aus, und halte an, genau bevor die leere Kachel sich das erste mal von der Eingangsumgebung ins Zentrum bewegt. Bezeichne die Konfiguration als C_1 . Führe LSG weiter aus, und halte wieder an, genau bevor die leere Kachel in die Umgebung des Kopfknotens wandert. Bezeichne die Konfiguration mit C_2 und die Untersequenz von LSG zwischen C_1 und C_2 mit S . Laut Lemma 2 und Lemma 3 kann die leere Kachel aus Konfiguration C_2 nicht zurück ins Zentrum wandern. Daher muss das Zentrum in C_1 dem in B_1 entsprechen und das in C_2 dem in B_2 . Nach Definition müssen $c+1$ neue Kacheln ins Zentrum gewandert sein, und $c+1$ vorhandene Kacheln aus ihm herausgerutscht sein. Nach Lemma 1 und der Definition von S kann die leere Kachel die Kantenumgebung in LSG niemals verlassen. Da genauso viele Kacheln in die Kante wandern müssen, wie aus ihr heraus kommen, muss die leere Kachel mit genau c anderen Kacheln in S durch das Zentrum der Kante wandern. \square

Es wurde jetzt gezeigt, dass jede Kante genau einmal mit der richtigen Anzahl an Elementkacheln durchlaufen wird. Daher kann die Sequenz LSG in eine Sequenz übersetzt werden, die Kanten in einem Graph traversiert und damit eine Lösung für REL liefert, wofür bereits gezeigt wurde, wie man eine gültige Belegung für 2/2/4-SAT erzeugt.

Damit ist der Beweis abgeschlossen und nPUZ ist NP-Vollständig