Hochschule RheinMain
Fachbereich Design Informatik Medien
Studiengang Master Informatik

Master-Arbeit
zur Erlangung des akademischen Grades
Master of Science - M.Sc.

Komplexitätstheoretische Betrachtungen von gitterbasierten Kryptografiesystemen

vorgelegt von Patrick Vogt

am 17. Juli 2013

Referent: Prof. Dr. Steffen Reith
Korreferent: Prof. Dr. Bodo Igler

Erklärung gem. ABPO, Ziff. 6.4.3

Ich versichere, dass ich die Master-Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Wiesbaden, 17. Juli 2013 Patrick Vogt

Hiermit erkläre ich mein Einverständnis mit den im Folgenden aufgeführten Verbreitungsformen dieser Master-Arbeit:

Verbreitungsform	**ja**	**nein**
Einstellung der Arbeit in die Hochschulbibliothek (mit Datenträger)		×
Einstellung der Arbeit in die Hochschulbibliothek (ohne Datenträger)	×	
Veröffentlichung des Titels der Arbeit im Internet	×	
Veröffentlichung der Arbeit im Internet	×	

Wiesbaden, 17. Juli 2013 Patrick Vogt

Inhaltsverzeichnis

Zusammenfassung

Viele asymmetrische Kryptografieverfahren basieren auf „schwierigen“ Problemen der Zahlentheorie, wie z. B. das Problem der Primfaktorzerlegung (FACTOR) oder das Problem des diskreten Logarithmus (DLOG). Das Lösen des Faktorisierungsproblems in polynomieller Zeit würde bspw. das RSA-Kryptografieverfahren unsicher bzw. unbrauchbar machen [WP04, S. 1]. Der Algorithmus von Shor [Sho97] zeigt, dass die beiden bereits erwähnten Probleme FACTOR und DLOG theoretisch mithilfe eines Quantencomputers in polynomieller Zeit gelöst werden können. Sollten technisch leistungsfähige Quantencomputer mit ausreichend vielen Quantum-Bits zur Verfügung stehen, so muss es andere Berechnungsprobleme bzw. darauf basierende Kryptografieverfahren geben, die auch mithilfe von Quantencomputer noch „schwierig“ zu lösen sind [WP04, S. 1]. Aus der Vielzahl dieser möglichen Berechnungsprobleme werden zwei Beispiele aus dem Bereich der gitterbasierten Kryptografie ausgewählt, die analysiert werden. Neben der Vorstellung dieser jeweiligen Berechnungsprobleme werden Algorithmen, die diese Probleme approximativ lösen können und komplexitätstheoretische Resultate der jeweiligen Probleme angegeben. Des Weiteren wird beschrieben, wie mithilfe dieser Berechnungsprobleme Informationen ver- bzw. entschlüsselt oder eine Nachricht signiert werden kann.

Kapitel 1

Einleitung

In diesem Kapitel werden die wichtigsten Begriffe eingeführt, die zum Verständnis von asymmetrischen Kryptografieverfahren und der sogenannten Post-Quantum Kryptografie von Bedeutung sind. Es wird zusätzlich eine kurze Motivation für das Thema der Post-Quatum Kryptografie gegeben und erläutert, warum dieses Themengebiet schon heute erforscht werden muss.

1.1 Asymmetrische Kryptografie

Asymmetrische Kryptografieverfahren werden bspw. zum sicheren Austausch von kryptografischen *Schlüsseln* zwischen unbekannten Kommunikationspartnern im Internet benötigt. Im Gegensatz zu den symmetrischen Kryptografieverfahren, die zum Ver- und Entschlüsseln den gleichen Schlüssel voraussetzen, wird bei asymmetrischen Kryptografieverfahren ein Schlüsselpaar benötigt. Das Schlüsselpaar besteht aus einem öffentlichen und einem zugehörigen privaten Schlüssel, die sich unterscheiden. Der öffentliche Schlüssel kann hierbei an einer zentralen öffentlichen Stelle hinterlegt werden, wohingegen der private Schlüssel geheim gehalten werden muss. Soll einem Kommunikationspartner eine geheime Nachricht geschickt werden, so wird die Nachricht mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und an den Empfänger gesendet. Diese verschlüsselte Nachricht kann dann nur vom Empfänger mithilfe seines privaten Schlüssels entschlüsselt und gelesen werden. Dieser Prozess wird durch Abbildung 1.1 verdeutlicht. In dieser Abbildung übernimmt „Alice“ die Rolle des Senders und „Bob“ die Rolle des Empfängers.

Damit ein solches asymmetrisches Kryptografieverfahren sicher ist, muss garantiert sein, dass aus dem öffentlichen Schlüssel nicht der private Schlüssel abgeleitet werden kann. Zu diesem Zweck wird für die Konstruktion eines asymmetrischen Kryptografieverfahrens eine Einwegfunktion benötigt.

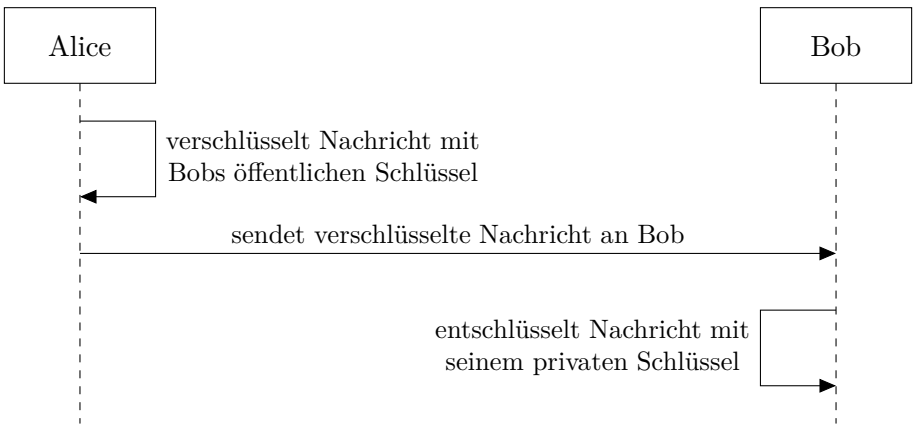


Abbildung 1.1: Die Ver- und Entschlüsselung mithilfe eines asymmetrischen Kryptografieverfahrens.

Eine Einwegfunktion ist eine Funktion f, die aus einem gegebenen Urbild x effizient das Bild $f(x)$ berechnen kann. Die Umkehrfunktion f^{-1}, das Zurückrechnen des Urbilds x aus einem gegebenen Bild $f(x)$, soll dabei praktisch unmöglich sein. Solche Einwegfunktionen können nur existieren, wenn für die beiden Komplexitätsklassen NP und BPP der Zusammenhang $\mathsf{NP} \not\subseteq \mathsf{BPP}$ gilt [Gol04, S. 27]. Diese Vermutung konnte bisher weder bewiesen noch widerlegt werden.

Es werden zwei Beispiele für Einwegfunktionen gegeben. Ein anschauliches Beispiel ist die Suche in einem Telefonbuch: Zu einem gegebenen Namen ist es einfach, die zugehörige Telefonnummer herauszusuchen. Die Umkehrung, das Finden eines Namens zu einer gegebenen Telefonnummer, ist dagegen schwierig. Das zweite Beispiel stammt aus dem Bereich der Zahlentheorie: Seien p und q zwei beliebige Primzahlen. Das Produkt $p \cdot q$ dieser beiden Zahlen ist einfach zu berechnen. Die Umkehrung dieser Funktion, das Faktorisieren des Produkts $p \cdot q$ in die beiden Primteiler p und q, ist dagegen, für hinreichend große Primzahlen p und q, praktisch unmöglich. Dieses Problem wird auch als Faktorisierungsproblem bezeichnet.

Damit eine solche Einwegfunktion für ein Kryptografieverfahren benutzt werden kann, muss diese eine weitere Eigenschaft, eine sogenannte *Falltür*, besitzen. Eine *Falltür* ermöglicht es, mithilfe einer geheimen Information, effizient aus dem gegebenen Bild $f(x)$ das entsprechende Urbild x zu berechnen. Im Falle von asymmetrischen Kryptografieverfahren ist diese geheime Information der private Schlüssel.

Es existieren mittlerweile mehrere Beispiele für asymmetrische Kryptografieverfahren. Das erste asymmetrische Kryptografieverfahren war RSA, das 1978 mithilfe des Faktorisierungsproblems konstruiert wurde [RSA78][Buc09, S. 137]. Neben der Verschlüsselung von Nachrichten bieten asymmetrische Kryptografieverfahren die Möglichkeit zum Signieren von Nachrichten oder die Identifikation eines unbekannten Kommunikationspartners.

1.2 Post-Quantum Kryptografie

Viele asymmetrische Kryptografieverfahren basieren auf „schwierigen“ Problemen der Zahlentheorie. Das Lösen des Faktorisierungsproblems in polynomieller Zeit würde bspw. das bereits erwähnte RSA-Kryptografieverfahren unsicher bzw. unbrauchbar machen [WP04, S. 1]. Der Algorithmus von Shor [Sho97] zeigt, dass das Faktorisierungsproblem mithilfe eines Quantencomputers in polynomieller Zeit gelöst werden kann. Sollten technisch leistungsfähige Quantencomputer mit ausreichend vielen Quantum-Bits zur Verfügung stehen, so muss es andere Berechnungsprobleme bzw. darauf basierende Kryptografieverfahren geben, die auch mithilfe von Quantencomputern noch „schwierig“ zu lösen sind [WP04, S. 1]. Laut [BB09, S. 17] besteht wenig Hoffnung, dass solche quantenresistente Berechnungsprobleme im Bereich der Zahlentheorie existieren.

Das Themengebiet, das sich mit der Suche und der Analyse von solchen quantenresistenten Kryptografieverfahren beschäftigt, wird als Post-Quantum Kryptografie bezeichnet. Dabei handelt es sich um kryptografische Verfahren, von denen angenommen wird, dass sie gegen *Angriffe* eines Quantencomputers, bei einer geeigneten Schlüssellänge, ausreichend abgesichert sind. Diese Verfahren setzen zum Ver- und Entschlüsseln keinen Quantencomputer voraus, sondern können mit klassischen Computern durchgeführt werden. Kryptografische Verfahren, die zum Ver- und Entschlüsseln einen Quantencomputer voraussetzen, werden mit dem Begriff der Quantenkryptografie bezeichnet.

Die Post-Quantum Kryptografie beinhaltet u. a. folgende asymmetrische Kryptografieverfahren, die als sicher gegen *Angriffe* von Quantencomputern gelten (aus [BB09, S. 1 f.]):

- Hash-basierte Kryptografie
- Codebasierte Kryptografie
- Gitterbasierte Kryptografie

Laut [BB09, S. 6] beeinflussen Quantencomputer die Sicherheit von symmetrischen Kryptografieverfahren nur in geringem Maße. Bei diesen Verfahren besteht das Problem, das zwischen den meist unbekannten Kommunikationspartnern zuerst ein geheimer Schlüssel über einen öffentlichen Kanal ausgetauscht werden muss. Die aufgelisteten asymmetrischen Kryptografieverfahren können derzeit noch nicht benutzt werden, da die Effizienz dieser Verfahren (noch) nicht für die praktische Anwendung ausreicht. Des Weiteren sind die Schlüsselgrößen bei diesen Verfahren (noch) zu groß, um etablierte Kryptografieverfahren, wie z. B. RSA, ablösen zu können. Diese Verfahren gelten (noch) als sicher, da bis jetzt kein (Quanten-)Algorithmus bekannt ist, der diese Kryptografieverfahren effizient angreifbar macht. Es ist somit wichtig, dass nach einer Effizienzsteigerung dieser Verfahren geforscht wird, um diese Verfahren in der Praxis einsetzen zu können. Die ständige

Analyse der oben beschriebenen Kryptografieverfahren ist genauso wichtig, damit davon ausgegangen werden kann, dass die entsprechenden Verfahren auch praktisch sicher sind [BB09, S. 2 f. und S. 11].

Die Suche nach einem effizienten und sicheren Kandidaten der Post-Quantum Kryptografie muss aus mehreren Gründen schon heute vorangetrieben werden: Die geheimen Daten, die heutzutage mit dem RSA-Verfahren verschlüsselt werden, können mithilfe von technisch leistungsfähigen Quantencomputern entschlüsselt werden, indem das Faktorisierungsproblem gelöst wird. Daten, die trotz leistungsfähiger Quantencomputer nicht entschlüsselt werden sollen, müssen bereits heute mit einem quantenresistenten Kryptografieverfahren verschlüsselt werden [BB09, S. 15].

Aus den aktuell als sicher geltenden Kryptografieverfahren wurde eine Menge von Verfahren ausgewählt, die innerhalb dieser Master-Arbeit genauer analyisert werden: Die gitterbasierte Kryptografie.

Kapitel 2

Mathematische Grundlagen

In diesem Kapitel werden die wichtigsten mathematischen Grundlagen für die gitterbasierten Kryptografieverfahren eingeführt. Dabei werden einige Kenntnisse aus der Algebra und der Linearen Algebra als bekannt vorausgesetzt. Die Bezeichnung der einzelnen mathematischen Begriffe wird meistens aus den englischen Begriffen abgeleitet. So wird im Folgenden ein Gitter mit dem Buchstaben $\mathcal{L}$ für „*Lattice*“ oder eine Grundmasche mit dem Buchstaben $\mathcal{P}$ für „*Parallelepiped*“ bezeichnet. Auf eine Angabe der englischen Begriffe wird im folgenden Text verzichtet.

Die mathematischen Beweise, die in diesem Kapitel angegeben sind, basieren meistens auf [MG02]. Die Ideen und die einzelnen Schritte dieser Beweise werden ausführlich beschrieben, sodass weniger Kenntnisse zum Verständnis dieser Beweise vorausgesetzt werden.

2.1 Einführung in die Gittertheorie

Als Erstes wird das Konzept der Gitter eingeführt, welche die Grundlage der gitterbasierten Kryptografieverfahren darstellen. Die vorgestellten Eigenschaften der Gitter werden entweder direkt bewiesen oder es wird eine Referenz auf Literatur angegeben, in der sich ein entsprechender Beweis befindet.

Definition 2.1 *Ein Gitter* $\mathcal{L}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n) \subset \mathbb{R}^m$ *ist eine Menge von Vektoren und wird durch* n *linear unabhängige Vektoren* $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{R}^m$ *erzeugt, wobei* $m, n \in \mathbb{N}$ *und* $1 < n \leq m$ *gilt. Dabei ist* n *der* Rang *des Gitters und* m *die* Dimension *des Gitters. Die* m*-dimensionalen Vektoren* $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ *werden dabei auch als* Basisvektoren des Gitters $\mathcal{L}$ *bzw. kurz als* Basisvektoren *bezeichnet. Gilt* $m = n$*, so wird das Gitter als* vollständiges Gitter *bezeichnet.*

Die Vektoren des Gitters $\mathcal{L}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$, die durch die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{R}^m$ erzeugt werden, sind wie folgt definiert:

$$\mathcal{L}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n) =_{\text{def}} \{x_1\vec{b}_1 + x_2\vec{b}_2 + \cdots + x_n\vec{b}_n\}, \text{ wobei } x_1, x_2, \ldots, x_n \in \mathbb{Z} \text{ ist.}$$

Jeder Vektor innerhalb eines Gitters kann als ganzzahlige Linearkombination der Basisvektoren beschrieben werden.

Anmerkung: Anstatt der Schreibweise $\mathcal{L}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$ für ein Gitter $\mathcal{L}$, das durch die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ erzeugt wird, wird auch die Schreibweise $\mathcal{L}(B)$ verwendet. Dabei ist

$$B =_{\text{def}} \left[\vec{b}_1 \mid \vec{b}_2 \mid \cdots \mid \vec{b}_n\right] \in \mathbb{R}^{m \times n} \text{ mit } m \geq n$$

eine Matrix, welche die n Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{R}^m$ als Spaltenvektoren besitzt. Diese Matrix wird als *Basis des Gitters* $\mathcal{L}$ oder kurz als *Gitterbasis* bezeichnet.

Ein Gittervektor entspricht in dieser Schreibweise der folgenden Menge von Vektoren:

$$\mathcal{L}(B) = \{B\vec{x} : \vec{x} \in \mathbb{Z}^n\}.$$

Im Folgenden wird zwischen diesen beiden Schreibweisen für eine Gitterbasis nach Anwendungsfall gewechselt.

Die Abbildung 2.1 veranschaulicht ein Gitter für $m = n = 2$ und zwei mögliche Basen für dieses Gitter. Sobald die mathematischen Grundlagen erläutert wurden, zeigt ein späteres Beispiel, dass die beiden Gitterbasen in Abbildung 2.1 ein äquivalentes Gitter erzeugen.

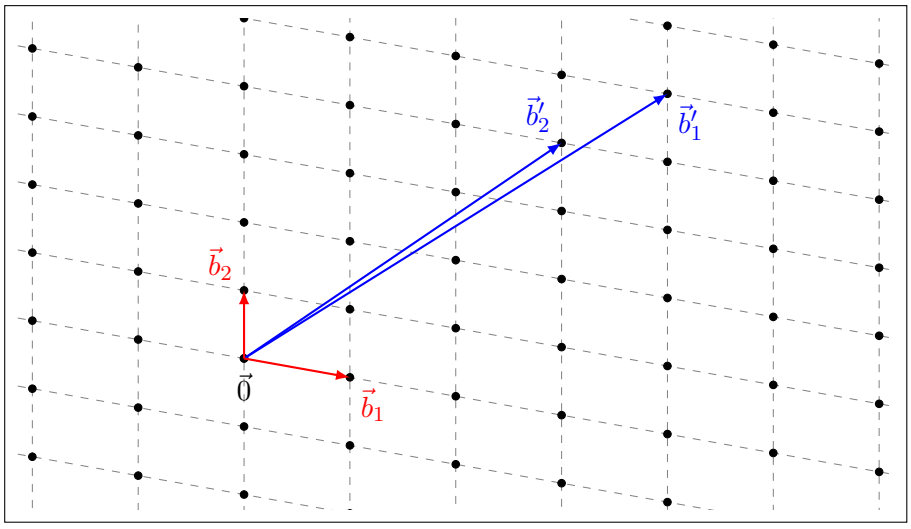


Abbildung 2.1: Ein zweidimensionales Gitter und zwei Gitterbasen (nach [BB09, S. 148]).

Definition 2.2 *Sei $\mathcal{L}(B) \subset \mathbb{R}^m$ ein Gitter, so ist die Menge*

$$\operatorname{span}(\mathcal{L}(B)) = \operatorname{span}(B) =_{\text{def}} \{B\vec{q} : \vec{q} \in \mathbb{R}^n\}.$$

definiert als lineare Hülle *des Gitters. Wenn $\mathcal{L}(B)$ ein vollständiges Gitter mit $m = n$ ist, so gilt* $\operatorname{span}(B) = \mathbb{R}^m$.

Eine äquivalente Definition für Gitter, die für bestimmte Eigenschaften oder Beweise besser geeignet ist, ist die folgende:

Definition 2.3 *(nach [MG02, S. 5] und [Gal12, S. 339])*

Ein Gitter Λ ist eine diskrete *nichtleere Teilmenge von $\mathbb{R}^m$, die bezüglich der Subtraktion abgeschlossen ist. Daraus folgt, dass wenn der Vektor $\vec{v} \in \Lambda$ und der Vektor $\vec{w} \in \Lambda$, auch der Differenzvektor $\vec{v} - \vec{w} \in \Lambda$ und somit im Gitter Λ enthalten ist.*

Diskret *bedeutet, dass für jede reelle Zahl $r > 0$ die Menge $\{\vec{v} \in \Lambda : \|\vec{v}\| \leq r\}$ endlich ist. Dabei ist $\|\vec{v}\|$ die Länge des Gittervektors $\vec{v}$ bezüglich einer beliebigen Norm.*

Anmerkung: Die Bezeichnung Λ für die Sichtweise eines Gitters als diskrete nichtleere Teilmenge von $\mathbb{R}^m$ ohne Angabe einer Gitterbasis wird gewählt, damit diese leicht von der Bezeichnung $\mathcal{L}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$ bzw. $\mathcal{L}(B)$ unterschieden werden kann.

Satz 2.4 *Sei Λ ein Gitter, eine diskrete nichtleere Teilmenge von $\mathbb{R}^m$, die bezüglich der Subtraktion abgeschlossen ist. Daraus folgt, dass Λ gemeinsam mit der Vektoraddition eine* diskrete Untergruppe *von $\mathbb{R}^m$ ist.*

Beweis: (nach [MG02, S. 5])

Neutrales Element: Das neutrale Element der Gruppe $(\mathbb{R}^m, +)$, der Nullvektor $\vec{0} \in \mathbb{R}^m$, ist in Λ enthalten, da $\vec{0} = \vec{v} - \vec{v} \in \Lambda$ gilt.

Inverses Element: Für jeden Gittervektor $\vec{v} \in \Lambda$ ist auch das inverse Element $-\vec{v} = \vec{0} - \vec{v} \in \Lambda$ im Gitter enthalten.

Abgeschlossenheit: Für zwei beliebige Gittervektoren $\vec{v} \in \Lambda$ und $\vec{w} \in \Lambda$ gilt, dass auch $\vec{v} + \vec{w} = \vec{v} - (-\vec{w}) \in \Lambda$ im Gitter enthalten ist.

□

Definition 2.5 *Sei $U \in \mathbb{Z}^{n \times n}$ eine ganzzahlige quadratische Matrix mit der Eigenschaft, dass $|\det(U)| = 1$ ist, so wird diese Matrix U auch ganzzahlige* unimodulare Matrix *genannt.*

Mithilfe von ganzzahligen unimodularen Matrizen können Gitterbasen transformiert werden, ohne dass das erzeugte Gitter verändert wird. Zuerst wird gezeigt, dass die Menge der ganzzahligen quadratischen unimodularen Matrizen eine Untergruppe der allgemeinen linearen Gruppe ist.

Proposition 2.6 *(Determinantenproduktsatz [BSMM12, S. 286])*

Seien $M, M' \in \mathbb{R}^{n\times n}$ beliebige quadratische Matrizen, so gilt für die Determinanten der Matrizen M, M' und der Determinante vom Matrixprodukt $M \cdot M'$ folgender Zusammenhang:

$$|\det(M)| \cdot |\det(M')| = |\det(M) \cdot \det(M')| = |\det(M \cdot M')|\,.$$

Proposition 2.7 *(Berechnung der inversen Matrix [BSMM12, S. 285])*

Die inverse Matrix M^{-1} kann mithilfe der folgenden Formel berechnet werden:

$$M^{-1} = \frac{M_{\text{adj}}}{\det(M)}\,.$$

Dabei ist M_{adj} die Adjunkte der Matrix M, die mithilfe von Streichungsmatrizen und Unterdeterminanten berechnet wird.

Hilfssatz 2.8 *Die Menge aller ganzzahligen unimodularen Matrizen $U \in \mathbb{Z}^{n\times n}$ gemeinsam mit der Matrixmultiplikation bildet eine Untergruppe der allgemeinen linearen Gruppe.*

Beweis:

Abgeschlossenheit: Das Ergebnis der Matrixmultiplikation zweier beliebiger Matrizen $U, U' \in \mathbb{Z}^{n\times n}$ ergibt wiederum ein Element von $\mathbb{Z}^{n\times n}$, da bei der Matrixmultiplikation nur addiert bzw. subtrahiert und multipliziert wird. Diese Verknüpfungen sind bzgl. der Menge der ganzen Zahlen $\mathbb{Z}$ abgeschlossen.

Mithilfe von Proposition 2.6 gilt für die Verknüpfung zweier ganzzahliger quadratischer unimodularer Matrizen:

$$|\det(U)| \cdot |\det(U')| = |\det(U \cdot U')|\,.$$

Da die Determinanten der Matrizen U und U' laut dem zu beweisenden Hilfssatz unimodular sind und somit einen Absolutwert von 1 besitzen, gilt:

$$|\det(U)| \cdot |\det(U')| = 1 \cdot 1 = 1 = |\det(U \cdot U')|\,.$$

Somit ist die Verknüpfung zweier ganzzahliger quadratischer unimodularer Matrizen wiederum eine ganzzahlige quadratische unimodulare Matrix.

Neutrales Element: Das neutrale Element der allgemeinen linearen Gruppe, die quadratische n-dimensionale Identitätsmatrix $I_n \in \mathbb{Z}^{n\times n}$, ist auch das neutrale Element der Untergruppe, denn für jede Matrix $U \in \mathbb{Z}^{n\times n}$ gilt:

$$U \cdot I_n = I_n \cdot U = U\,.$$

Des Weiteren besitzt die Einheitsmatrix I_n die Eigenschaft, dass $|\det(I_n)| = 1$ ist und somit ist I_n in der Menge der ganzzahligen quadratischen unimodularen Matrizen enthalten.

Inverses Element: Zu jeder unimodularen Matrix $U \in \mathbb{Z}^{n\times n}$ existiert eine inverse Matrix $U^{-1} \in \mathbb{Z}^{n\times n}$, sodass $U \cdot U^{-1} = I_n$ gilt.

Als Erstes muss gezeigt werden, dass die inverse Matrix U^{-1} nur Matrixelemente aus den ganzen Zahlen $\mathbb{Z}$ enthält. Diese Eigenschaft ergibt sich aus Proposition 2.7:

$$U^{-1} = \frac{U_{\text{adj}}}{\det(U)}\,.$$

Da $U \in \mathbb{Z}^{n\times n}$ ist, gilt auch für die Adjunkte U_{adj} der Matrix U die Eigenschaft $U_{\text{adj}} \in \mathbb{Z}^{n\times n}$. Da $\det(U) \in \{-1, 1\}$ ist, muss auch die inverse Matrix $U^{-1} \in \mathbb{Z}^{n\times n}$ sein.

Es bleibt zu zeigen, dass die inverse Matrix U^{-1} unimodular ist. Aus der Gleichung

$$U \cdot U^{-1} = I_n$$

ergibt sich mit Proposition 2.6:

$$\begin{aligned} |\det(U \cdot U^{-1})| &= |\det(I_n)| \\ |\det(U)| \cdot |\det(U^{-1})| &= |\det(I_n)|\,. \end{aligned}$$

Da die Matrizen U und I_n unimodular sind, gilt:

$$\begin{aligned} 1 \cdot |\det(U^{-1})| &= 1 \\ |\det(U^{-1})| &= 1\,. \end{aligned}$$

Somit ist die inverse Matrix U^{-1} ganzzahlig, quadratisch und unimodular.

□

Es wird noch eine Eigenschaft von ganzzahligen quadratischen unimodularen Matrizen hervorgehoben, die in den folgenden Beweisen benutzt wird.

Hilfssatz 2.9 *Seien $U, U^{-1} \in \mathbb{Z}^{n\times n}$ zwei ganzzahlige quadratische Matrizen, sodass $I_n = U \cdot U^{-1}$ gilt, dann folgt daraus, dass $|det(U)| = 1$ sein muss.*

Beweis:

Aus $U, U^{-1} \in \mathbb{Z}^{n\times n}$ folgt:

$$\det(U) \in \mathbb{Z}\,, \text{ sowie } \det(U^{-1}) \in \mathbb{Z}\,.$$

Aus $I_n = U \cdot U^{-1}$ folgt weiter:

$$\begin{aligned} 1 = \det(I_n) = \det(U \cdot U^{-1}) &= \det(U) \cdot \det(U^{-1}) \\ 1 &= \det(U) \cdot \det(U^{-1})\,. \end{aligned}$$

Zusammen: Sowohl $|\det(U)| = 1$, als auch $|\det(U^{-1})| = 1$ müssen gelten.

□

Die folgenden Sätze zeigen, wie Gitterbasen bzw. die dadurch erzeugten Gitter in Beziehung zueinander stehen können. Hierfür wird u. a. die Menge der ganzzahligen quadratischen unimodularen Matrizen benötigt.

Hilfssatz 2.10 *Seien $\mathcal{L}(B) \subset \mathbb{R}^m$ und $\mathcal{L}(B') \subset \mathbb{R}^m$ zwei Gitter. Gilt für alle Basisvektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n \in \mathbb{R}^m$ des Gitters $\mathcal{L}(B')$, dass diese im Gitter $\mathcal{L}(B)$ enthalten sind, so ist $\mathcal{L}(B') \subseteq \mathcal{L}(B)$. Das Gitter $\mathcal{L}(B')$ wird auch* Teilgitter *von $\mathcal{L}(B)$ genannt.*

Beweis:

Da alle Basisvektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n$ des Gitters $\mathcal{L}(B')$ auch Vektoren im Gitter $\mathcal{L}(B)$ sind, so sind nach Definition der Gittervektoren auch alle ganzzahligen Linearkombinationen dieser Vektoren im Gitter $\mathcal{L}(B)$ enthalten. Alle ganzzahlige Linearkombinationen der Vektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n$ entspricht der Menge aller Gittervektoren von $\mathcal{L}(B')$ und somit gilt, dass das Gitter $\mathcal{L}(B')$ komplett im Gitter $\mathcal{L}(B)$ enthalten ist. Daraus folgt $\mathcal{L}(B') \subseteq \mathcal{L}(B)$.

□

Folgerung 2.11 *Seien $\mathcal{L}(B) \subset \mathbb{R}^m$ und $\mathcal{L}(B') \subset \mathbb{R}^m$ zwei beliebige Gitter. Gilt $\mathcal{L}(B') \subseteq \mathcal{L}(B)$ und zustäzlich $\mathcal{L}(B) \subseteq \mathcal{L}(B')$, so gilt $\mathcal{L}(B) = \mathcal{L}(B')$ und die beiden Gitter $\mathcal{L}(B')$ und $\mathcal{L}(B)$ sind äquivalent.*

Satz 2.12 *Die Basen $B, B' \in \mathbb{R}^{m\times n}$ mit $m \geq n$ erzeugen äquivalente Gitter $\mathcal{L}(B) \subset \mathbb{R}^m$ bzw. $\mathcal{L}(B') \subset \mathbb{R}^m$ genau dann, wenn eine unimodulare Matrix $U \in \mathbb{Z}^{n\times n}$ existiert, sodass $B' = B \cdot U$ gilt.*

Beweis:

„⇒“: Da die beiden Basen B, B' das gleiche Gitter erzeugen, müssen die Basisvektoren der Basis B' ganzzahlige Linearkombinationen der Basisvektoren der Basis B sein. Dies kann durch folgendes lineares Gleichungssystem dargestellt werden:

$$B' = B \cdot U \text{ , wobei } U \in \mathbb{Z}^{n \times n} \text{ ist.}$$

Da B bzw. B' eine Basis von $\mathcal{L}$ ist und somit alle Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ bzw. $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n$ linear unabhängig sind, muss $\det(U) \neq 0$ gelten. Aus dieser Eigenschaft folgt, dass zu der Matrix $U \in \mathbb{Z}^{n \times n}$ eine inverse Matrix U^{-1} existiert, sodass $B' \cdot U^{-1} = B$ ist. Da die Basisvektoren der Basis B wiederum ganzzahlige Linearkombinationen der Basisvektoren der Basis B' sind, muss auch $U^{-1} \in \mathbb{Z}^{n \times n}$ gelten. Aus Hilfssatz 2.9 ergibt sich, dass die Matrix U somit unimodular sein muss.

„⇐“: Da $B' = B \cdot U$ und $U \in \mathbb{Z}^{n \times n}$ gilt, folgt daraus, dass $\mathcal{L}(B') \subseteq \mathcal{L}(B)$ ist. Da die Matrix U unimodular ist, folgt daraus, dass eine inverse unimodulare Matrix $U^{-1} \in \mathbb{Z}^{n \times n}$ existiert und somit folgt aus $B' = B \cdot U$, dass auch $B' \cdot U^{-1} = B$ gilt. Daraus folgt, dass $\mathcal{L}(B) \subseteq \mathcal{L}(B')$ ist. Zusammen: $\mathcal{L}(B) = \mathcal{L}(B')$.

□

Folgerung 2.13 *Erzeugen die Basen $B, B' \in \mathbb{R}^{m \times n}$ mit $m \geq n$ äquivalente Gitter $\mathcal{L}(B) = \mathcal{L}(B')$, so gilt für die lineare Hüllen* $\text{span}(B)$ *und* $\text{span}(B')$ *folgender Zusammenhang:*

$$\text{span}(B) = \text{span}(B') \,.$$

Die linearen Hüllen zweier äquivalenter Gitter $\mathcal{L}(B)$ und $\mathcal{L}(B')$ sind demnach gleich.

Anmerkung: Mithilfe von Satz 2.12 kann zum Analysieren eines Gitters $\mathcal{L}(B)$ eine beliebige Basis B benutzt werden, die das Gitter $\mathcal{L}$ erzeugt. Im Folgenden macht es keinen Unterschied, ob ein Gitter $\mathcal{L}$ mit $\mathcal{L}(B)$ oder $\mathcal{L}$ bezeichnet wird.

Das folgende Beispiel mit $m = n = 2$ zeigt, dass die beiden Gitterbasen in Abbildung 2.1 ein äquivalentes Gitter erzeugen.

Beispiel:

Gegeben seien die beiden Basen $B = \begin{pmatrix} 1.4 & 0.0 \\ -0.25 & 0.9 \end{pmatrix}$ und $B' = \begin{pmatrix} 5.6 & 4.2 \\ 3.5 & 2.85 \end{pmatrix}$.

Da beide Basen $B, B' \in \mathcal{L}$ sind, existiert eine Matrix $U \in \mathbb{Z}^{2\times 2}$, sodass $B' = B \cdot U$. Die Matrix U sei durch $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ gegeben. Aus $B' = B \cdot U$ folgt:

$$\begin{pmatrix} 5.6 & 4.2 \\ 3.5 & 2.85 \end{pmatrix} = \begin{pmatrix} 1.4 & 0.0 \\ -0.25 & 0.9 \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Es ergibt sich folgendes lineares Gleichungssystem:

$$\begin{aligned} 1.4 \cdot a &= 5.6 \\ 1.4 \cdot b &= 4.2 \\ -0.25 \cdot a + 0.9 \cdot c &= 3.5 \\ -0.25 \cdot b + 0.9 \cdot d &= 2.85\,. \end{aligned}$$

Daraus folgt, dass $U = \begin{pmatrix} 4 & 3 \\ 5 & 4 \end{pmatrix}$ und $|\det(U)| = 1$. Somit existiert die inverse Matrix $U^{-1} \in \mathbb{Z}^{2\times 2}$. Eine Berechnung der inversen Matrix führt zu $U^{-1} = \begin{pmatrix} 4 & -3 \\ -5 & 4 \end{pmatrix}$ und $|\det(U^{-1})| = 1$. Die beiden Gitter $\mathcal{L}(B)$ und $\mathcal{L}(B')$ sind äquivalent.

Es folgt ein zweites Beispiel, wiederum mit $m = n = 2$, in dem von zwei Gitterbasen unterschiedliche Gitter erzeugt werden.

Beispiel:
Gegeben seien die beiden Basen $B = \begin{pmatrix} 1.4 & 0.0 \\ -0.25 & 0.9 \end{pmatrix}$ und $B' = \begin{pmatrix} 1.4 & 0.0 \\ -0.25 & 1.8 \end{pmatrix}$.

Da beide Basen $B, B' \in \mathcal{L}$ sind, existiert eine Matrix $U \in \mathbb{Z}^{2\times 2}$, sodass $B' = B \cdot U$. Die Matrix U sei durch $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ gegeben. Aus $B' = B \cdot U$ folgt:

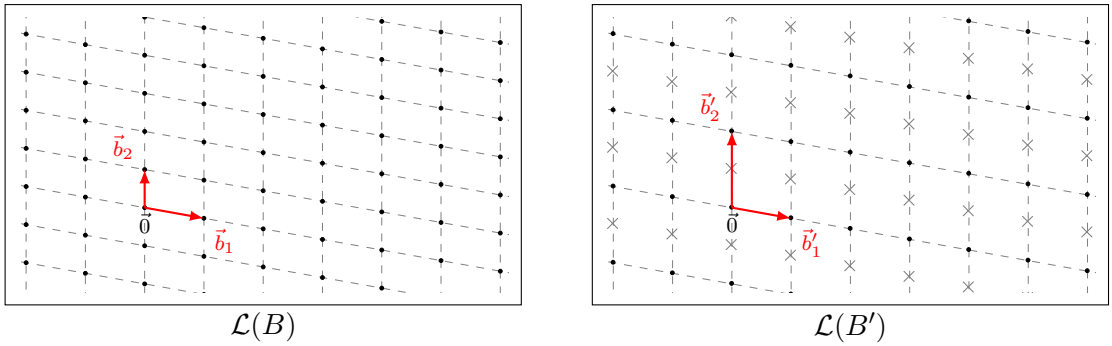
$$\begin{pmatrix} 1.4 & 0.0 \\ -0.25 & 1.8 \end{pmatrix} = \begin{pmatrix} 1.4 & 0.0 \\ -0.25 & 0.9 \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Es ergibt sich folgendes lineares Gleichungssystem:

$$\begin{aligned} 1.4 \cdot a &= 1.4 \\ 1.4 \cdot b &= 0.0 \\ -0.25 \cdot a + 0.9 \cdot c &= -0.25 \\ -0.25 \cdot b + 0.9 \cdot d &= 1.8\,. \end{aligned}$$

Daraus folgt, dass $U = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ und $|\det(U)| = 2$. Somit existiert die inverse Matrix U^{-1} zwar, diese ist jedoch nicht ganzzahlig, was durch $U^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$ ersichtlich ist.

Die beiden Basen $B, B' \in \mathcal{L}$ erzeugen unterschiedliche Gitter. Da die Basisvektoren der Basis B' im Gitter $\mathcal{L}(B)$ enthalten sind, folgt daraus, dass $\mathcal{L}(B')$ ein Teilgitter von $\mathcal{L}(B)$ ist. Die Abbildung 2.2 zeigt, dass die erzeugten Gitter nicht identisch sind.



$\mathcal{L}(B)$ $\mathcal{L}(B')$

Abbildung 2.2: Ein Vergleich zweier unterschiedlicher Gitter.

Folgerung 2.14 *Aus Satz 2.12 ergeben sich gültige Operationen, die eine Basis B eines Gitters $\mathcal{L}$ in eine Basis B' überführen können, ohne dass sich das durch die Basis B' erzeugte Gitter ändert. Es muss lediglich eine unimodulare Startmatrix $U \in \mathbb{Z}^{n \times n}$ gefunden werden, auf die alle Matrixoperationen durchgeführt werden können, welche die Determinante der Matrix nicht verändern bzw. nur einen Vorzeichenwechsel der Determinante vornehmen.*

Daraus ergeben sich folgende Matrixoperationen, die eine unimodulare Matrix U in eine unimodulare Matrix U' überführen, wobei $\vec{b}_i$ und $\vec{b}_j$ mit $i, j \in \{1, 2, \ldots, n\}$ und $i \neq j$ zwei beliebige Basisvektoren eines Gitters sind:

- *Es können zwei Spalten in der unimodularen Matrix U vertauscht werden. Hierdurch ändert sich das Vorzeichen der Determinante, aber die so erzeugte Matrix U' ist ebenfalls unimodular. Anschaulich bedeutet diese Operation, dass zwei beliebige Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ miteinander vertauscht werden.*

  Schreibweise: $\vec{b}_i \leftrightarrow \vec{b}_j$.

- *Eine Spalte in der unimodularen Matrix U kann negiert werden. Hierdurch ändert sich zwar das Vorzeichen der Determinante, aber die so erzeugte Matrix U' ist ebenfalls unimodular. Anschaulich bedeutet diese Operation, dass der Basisvektor $\vec{b}_i$ durch einen Vektor $-\vec{b}_i$ ersetzt wird. Dieser neue Basisvektor $-\vec{b}_i$ besitzt die gleiche Länge wie $\vec{b}_i$, zeigt jedoch in die entgegengesetzte Richtung.*

  $$\textit{Schreibweise: } \vec{b}_i \leftarrow -\vec{b}_i \,.$$

- *Zu einer Spalte in der unimodularen Matrix U kann das ganzzahlige Vielfache einer anderen Spalte hinzuaddiert bzw. subtrahiert werden. Die Determinante der so erzeugten Matrix U' ändert sich in diesem Fall nicht. Somit ist die Matrix U' ebenfalls unimodular. Anschaulich bedeutet dies, dass zu einem Basisvektor $\vec{b}_i$ das ganzzahlige Vielfache eines anderen Basisvektors $\vec{b}_j$ hinzuaddiert wird.*

  $$\textit{Schreibweise: } \vec{b}_i \leftarrow \vec{b}_i + k\vec{b}_j \text{ , wobei } k \in \mathbb{Z} \textit{ ist}.$$

Hinweis: Dies sind die Operationen des Gauß-Eliminationsverfahren. Da keine dieser Operationen die Unimodularität einer Matrix verändert, können diese Operationen beliebig hintereinander durchgeführt werden.

Des Weiteren sei verdeutlicht, dass nur $\vec{b}_i \leftarrow \vec{b}_i + k\vec{b}_j$ die Länge von Basisvektoren eines Gitters verändert. Bei $\vec{b}_i \leftrightarrow \vec{b}_j$ und $\vec{b}_i \leftarrow -\vec{b}_i$ wird die Länge der Basisvektoren nicht verändert.

Die „schwierigen“ Berechnungsprobleme in Gittern sind meistens von der Wahl der Gitterbasis abhängig. Es existieren „gute“ und „schlechte“ Basen. Diese Begriffe werden hier nicht weiter erläutert. Der folgende Satz zeigt, dass in einem Gitter mit einem hohen Rang n unendliche viele Basen existieren.

Satz 2.15 *Jedes Gitter $\mathcal{L}$ mit Rang $n > 1$ besitzt abzählbar unendlich viele Basen.*

Beweis:

Dieser Satz kann aus Folgerung 2.14 abgeleitet werden. Das Vertauschen und Negieren von Basisvektoren erzeugt eine endliche Menge von Gitterbasen. Das Hinzuaddieren eines vielfachen Basisvektors zu einem anderen Basisvektor erzeugt eine abzählbar unendliche Menge an unterschiedlichen Gitterbasen, da in der Zuweisung $\vec{b}'_i \leftarrow \vec{b}_i + k\vec{b}_j$ die Variable $k \in \mathbb{Z}$ und die Menge der ganzen Zahlen $\mathbb{Z}$ unendlich ist.

Bei einem Gitter mit Rang $n = 1$ besteht die Basis aus genau einem Basisvektor $\vec{b}_1$. Aus diesem Grund ist es nicht möglich, diesen Basisvektor mit einem anderen Basisvektor zu

vertauschen oder das ganzzahlige Vielfache eines anderen Basisvektors zu diesem Basisvektor hinzuzuaddieren. Die einzige mögliche Basistransformation ist das Negieren des Basisvektors $\vec{b}_1$, sodass $\vec{b}_1 \leftarrow -\vec{b}_1$ ist. Somit sind $\vec{b}_1$ und $-\vec{b}_1$ die einzigen beiden Basen eines eindimensionalen Gitters.

□

Die folgende Definition zeigt, dass es neben Teilgittern auch andere Beziehungen zwischen unterschiedlichen Gittern geben kann:

Definition 2.16 *Sei $\mathcal{L}(B)$ ein Gitter mit der Basis $B \in \mathbb{R}^{m\times n}$ mit $m \geq n$, dann wird die Menge:*

$$\mathcal{L}^*(B) =_{\text{def}} \{\vec{w} \in \text{span}(B) : \langle \vec{v}, \vec{w}\rangle \in \mathbb{Z} \text{ für alle } \vec{v} \in \mathcal{L}(B)\},$$

das duale Gitter *zu $\mathcal{L}(B)$ genannt, wobei $\langle \cdot, \cdot \rangle$ das Skalarprodukt angibt.*

Satz 2.17 *Sei $\mathcal{L}(B)$ ein Gitter mit der Basis $B \in \mathbb{R}^{m\times n}$ mit $m \geq n$, dann ist $\mathcal{L}^*(B) = \mathcal{L}(D)$ mit der Basis $D = B(B^TB)^{-1} \in \mathbb{R}^{m\times n}$ das duale Gitter zu $\mathcal{L}(B)$.*

Beweis:

Um zu zeigen, dass $\mathcal{L}^*(B) = \mathcal{L}(D)$ das duale Gitter zum Gitter $\mathcal{L}(B)$ ist, wird zuerst $\mathcal{L}(D) \subseteq \mathcal{L}^*(B)$ und danach $\mathcal{L}^*(B) \subseteq \mathcal{L}(D)$ gezeigt.

$\mathcal{L}(D) \subseteq \mathcal{L}^*(B)$: Sei $\vec{y} \in \mathbb{Z}^n$ und $D\vec{y} \in \mathbb{R}^m$ ein beliebiger Vektor im Gitter $\mathcal{L}(D)$, dann gelten folgende Eigenschaften:

$$\begin{aligned} D\vec{y} \in \text{span}(B), \text{ denn } D\vec{y} &= B \cdot (\underbrace{B^T}_{\in\mathbb{R}^{n\times m}} \cdot \overbrace{B}^{\in\mathbb{R}^{m\times n}})^{-1}\vec{y} = B\underbrace{(B^TB)^{-1}}_{\in\mathbb{R}^{n\times n}}\vec{y} \\ &= B\underbrace{(B^TB)^{-1}}_{\in\mathbb{R}^{n\times n}} \cdot \underbrace{\vec{y}}_{\in\mathbb{Z}^n} = B \cdot \underbrace{(B^TB)^{-1}\vec{y}}_{\in\mathbb{R}^n} = \underbrace{B(B^TB)^{-1}\vec{y}}_{\in\text{span}(B)}. \end{aligned}$$

Sei $B\vec{x} \in \mathcal{L}(B)$ und $D\vec{y} \in \mathcal{L}(D)$ mit $\vec{x}, \vec{y} \in \mathbb{Z}^n$, dann gilt für das Skalarprodukt der beiden Vektoren:

$$\begin{aligned} \langle B\vec{x}, D\vec{y}\rangle = \langle D\vec{y}, B\vec{x}\rangle &= (D\vec{y})^T(B\vec{x}) \overbrace{=}^{D=B(B^TB)^{-1}} (B(B^TB)^{-1}\vec{y})^T(B\vec{x}) = \vec{y}^T(B(B^TB)^{-1})^TB\vec{x} \\ &= \vec{y}^T((B^TB)^{-1})^TB^TB\vec{x} = \vec{y}^T((B^TB)^{-1})^T(B^TB)^T\vec{x} \\ &= \vec{y}^T(\underbrace{(B^TB)(B^TB)^{-1}}_{I_n})^T\vec{x} = \vec{y}^TI_n^T\vec{x} = \vec{y}^TI_n\vec{x} \\ &= \vec{y}^T\vec{x} = \langle \vec{y}, \vec{x}\rangle = \langle \vec{x}, \vec{y}\rangle \in \mathbb{Z}, \text{ da } \vec{x}, \vec{y} \in \mathbb{Z}^n. \end{aligned}$$

Daraus folgt, dass $\mathcal{L}(D) \subseteq \mathcal{L}^*(B)$ gilt.

$\mathcal{L}^*(B) \subseteq \mathcal{L}(D)$: Sei $\vec{u} \in \mathcal{L}^*(B)$ ein beliebiger Gittervektor im dualen Gitter von $\mathcal{L}(B)$, dann gelten die folgenden Eigenschaften:

$\vec{u} = B\vec{r}$ mit $\vec{r} \in \mathbb{R}^n$, da $\vec{u} \in \text{span}(B)$ ist und
$\vec{u} = B\vec{r} = BI_n\vec{r} = B((B^TB)^{-1}(B^TB))\vec{r} = B(B^TB)^{-1}B^TB\vec{r} = D(B^TB\vec{r}) = D\underbrace{(B^T\vec{u})}_{\in\mathbb{Z}^n}$.

Die letzte Folgerung, dass $(B^T\vec{u}) \in \mathbb{Z}^n$ ist, wird kurz verdeutlicht. Sei $B \in \mathbb{R}^{m\times n}$ die Matrix, die die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ als Spaltenvektoren beinhaltet, dann gilt für $(B^T\vec{u})$:

$$\begin{pmatrix} \vec{b}_1 & \vec{b}_2 & \cdots & \vec{b}_n \end{pmatrix}^T \cdot \vec{u} = \begin{pmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vdots \\ \vec{b}_n \end{pmatrix} \cdot \vec{u} = \begin{pmatrix} \langle \vec{b}_1, \vec{u} \rangle \\ \langle \vec{b}_2, \vec{u} \rangle \\ \vdots \\ \langle \vec{b}_n, \vec{u} \rangle \end{pmatrix} \in \mathbb{Z}^n,$$

da $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathcal{L}(B)$ und $\vec{u} \in \mathcal{L}^*(B)$ und nach Definition von dualen Gittern ein Skalarprodukt $\langle \vec{v}, \vec{w} \rangle \in \mathbb{Z}$ für alle $\vec{v} \in \mathcal{L}(B)$ und $\vec{w} \in \mathcal{L}^*(B)$ ist.

Daraus folgt schließlich: $D\underbrace{(B^T\vec{u})}_{\in\mathbb{Z}^n} = \underbrace{D(B^T\vec{u})}_{\in\mathcal{L}(D)}$ und somit $\mathcal{L}^*(B) \subseteq \mathcal{L}(D)$.

Da sowohl $\mathcal{L}(D) \subseteq \mathcal{L}^*(B)$, als auch $\mathcal{L}^*(B) \subseteq \mathcal{L}(D)$ gilt, muss $\mathcal{L}^*(B) = \mathcal{L}(D)$ gelten.

□

Folgerung 2.18 *Sei $\mathcal{L}(B)$ ein vollständiges Gitter mit der Dimension n und Basis $B \in \mathbb{R}^{n\times n}$, dann ist $\mathcal{L}^*(B) = \mathcal{L}(D)$ mit der Basis $D = (B^T)^{-1} \in \mathbb{R}^{n\times n}$ das vollständige duale Gitter zu $\mathcal{L}(B)$.*

Beweis:

Aus Satz 2.17 folgt, dass $\mathcal{L}^*(B) = \mathcal{L}(D)$ mit der Basis $D = B(B^TB)^{-1} \in \mathbb{R}^{m\times n}$ das duale Gitter zu $\mathcal{L}(B)$ erzeugt. Da das Gitter $\mathcal{L}(B)$ vollständig ist und somit $m = n$ gilt, ist die Basismatrix $B \in \mathbb{R}^{n\times n}$ quadratisch. Die Basis $D = B(B^TB)^{-1}$ des dualen Gitters kann dann wie folgt vereinfacht werden:

$$D = B(B^TB)^{-1} = \underbrace{BB^{-1}}_{I_n}(B^T)^{-1} = I_n(B^T)^{-1} = (B^T)^{-1}.$$

□

Folgerung 2.19 *Sei $\mathcal{L}(D)$ ein Gitter mit der Basis $D = B(B^TB)^{-1} \in \mathbb{R}^{m\times n}$ mit $m \geq n$, dann ist $\mathcal{L}^*(D) = \mathcal{L}(B)$ mit der Basis $B = D(D^TD)^{-1} \in \mathbb{R}^{m\times n}$ das duale Gitter zu $\mathcal{L}(D)$. Dies zeigt, dass die Dualität von Gitterbasen eine symmetrische Relation ist.*

Beweis:

Aus Satz 2.17 folgt, dass $\mathcal{L}^*(D)$ mit der Basis $D(D^TD)^{-1} \in \mathbb{R}^{m\times n}$ das duale Gitter zu $\mathcal{L}(D) = \mathcal{L}(B(B^TB)^{-1})$ erzeugt. Daraus folgt:

$$\begin{aligned}
D(D^TD)^{-1} &\underbrace{=}_{D=B(B^TB)^{-1}} B(B^TB)^{-1}(((B(B^TB)^{-1})^TB(B^TB)^{-1})^{-1} \\
&= B(B^TB)^{-1}((B^TB)^{-1})^TB^TB(B^TB)^{-1})^{-1} \\
&= B(B^TB)^{-1}((B^TB)^{-1})^T(B^TB)^T(B^TB)^{-1})^{-1} \\
&= B(B^TB)^{-1}(\underbrace{(B^TB)(B^TB)^{-1}}_{I_n})^T(B^TB)^{-1})^{-1} \\
&= B(B^TB)^{-1}(I_n^T(B^TB)^{-1})^{-1} \\
&= B(B^TB)^{-1}(I_n(B^TB)^{-1})^{-1} \\
&= B(B^TB)^{-1}\underbrace{((B^TB)^{-1})^{-1}}_{(B^TB)} \\
&= B\underbrace{(B^TB)^{-1}(B^TB)}_{I_n} \\
&= BI_n \\
&= B\,.
\end{aligned}$$

Dies zeigt, dass $\mathcal{L}(B)$ das duale Gitter zu $\mathcal{L}(D) = \mathcal{L}(B(B^TB)^{-1})$ ist.

□

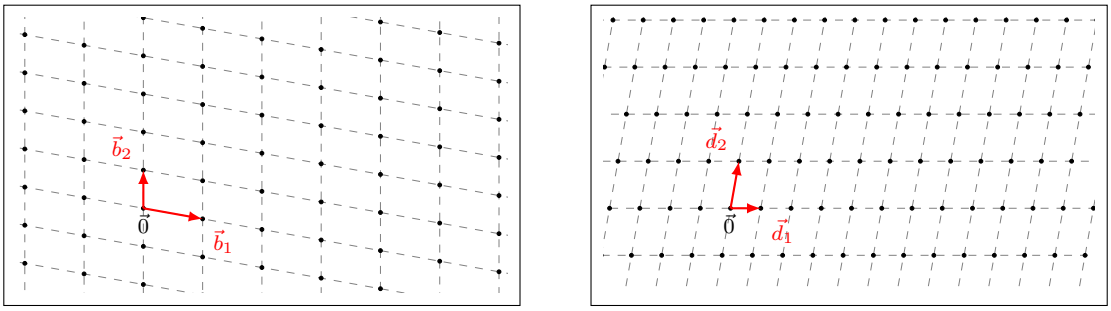
Die Abbildung 2.3 zeigt das duale Gitter $\mathcal{L}^*(B) = \mathcal{L}(D)$ zu dem Gitter $\mathcal{L}(B)$ in Abbildung 2.1.

2.2 Grundmasche eines Gitters

Neben der Menge der Gittervektoren existiert eine weitere Menge von Vektoren, die als Grundmasche des Gitters bezeichnet wird.

Definition 2.20 *Sei $\mathcal{L} \subset \mathbb{R}^m$ ein Gitter, das durch die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{R}^m$ erzeugt wird. Die Menge der Vektoren*

$$\mathcal{P}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n) = \mathcal{P}(B) =_{\text{def}} \{s_1\vec{b}_1 + s_2\vec{b}_2 + \cdots + s_n\vec{b}_n\}\text{, wobei } s_1, s_2, \ldots, s_n \in [0,1) \text{ ist,}$$



$\mathcal{L}(B)$ $\mathcal{L}^*(B) = \mathcal{L}(D)$

Abbildung 2.3: Ein Gitter $\mathcal{L}(B)$ und das zugehörige duale Gitter $\mathcal{L}^*(B) = \mathcal{L}(D)$.

wird als Grundmasche des Gitters $\mathcal{L}$ *bzw. kurz als* Grundmasche *bezeichnet.*

Die Abbildung 2.4 zeigt das zweidimensionale Gitter aus Abbildung 2.1 und die zur Basis zugehörige Grundmasche.

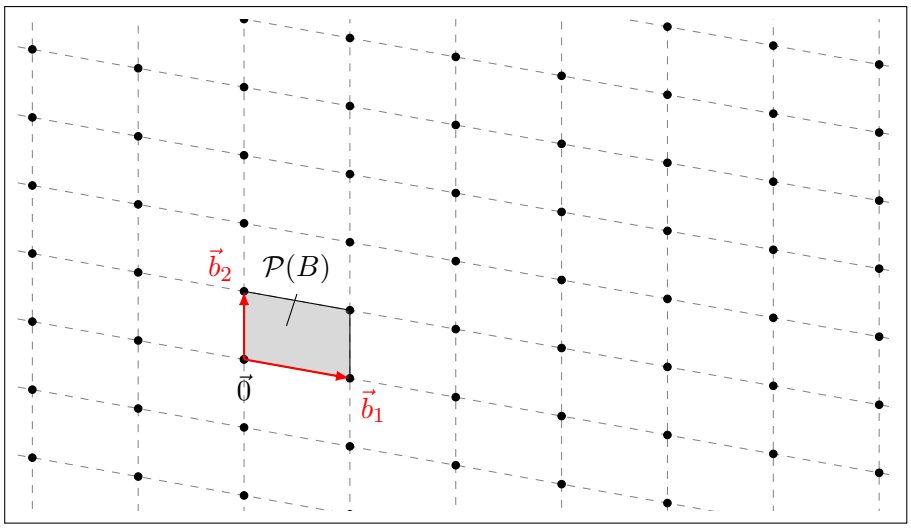


Abbildung 2.4: Ein zweidimensionales Gitter und die zur Basis zugehörige Grundmasche.

Die folgenden Sätze zeigen Eigenschaften, die das Gitter zusammen mit der Grundmasche besitzt.

Satz 2.21 *Sei $\mathcal{L}(B) \subset \mathbb{R}^m$ ein Gitter und $\mathcal{P}(B)$ die Grundmasche von $\mathcal{L}$, so ist jeder Vektor $\vec{q} \in \operatorname{span}(B)$ eindeutig durch zwei Vektoren $\vec{v} \in \mathcal{L}(B)$ und $\vec{\alpha} \in \mathcal{P}(B)$ bestimmt, sodass $\vec{q} = \vec{v} + \vec{\alpha}$ gilt.*

Beweis: (nach [HPS08, S. 366 f.])

Als Erstes muss die Zerlegung eines jeden Vektors $\vec{q} \in \operatorname{span}(B)$ in zwei Vektoren $\vec{v} \in \mathcal{L}(B)$ und $\vec{\alpha} \in \mathcal{P}(B)$ gezeigt werden. Da die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{R}^m$ linear

unabhängig sind, kann jeder Vektor $\vec{q} \in \text{span}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$ als reelle Linearkombination dieser Basisvektoren dargestellt werden:

$$\vec{q} = r_1\vec{b}_1 + r_2\vec{b}_2 + \cdots + r_n\vec{b}_n, \text{ wobei } r_1, r_2, \ldots, r_n \in \mathbb{R} \text{ ist}.$$

Die Koeffizienten $r_1, r_2, \ldots, r_n$ der Linearkombination von $\vec{q}$ können dabei in einen ganzzahligen Anteil $x_i \in \mathbb{Z}$ und einen reellen Anteil $s_i \in [0, 1)$ aufgetrennt werden, sodass $r_i = \lfloor r_i \rfloor + (r_i - \lfloor r_i \rfloor) = x_i + s_i$ ist, für $i \in \{1, 2, \ldots, n\}$.

Somit folgt daraus:

$$\begin{aligned}\vec{q} &= r_1\vec{b}_1 + r_2\vec{b}_2 + \cdots + r_n\vec{b}_n \\ &= (x_1 + s_1)\vec{b}_1 + (x_2 + s_2)\vec{b}_2 + \cdots + (x_n + s_n)\vec{b}_n\,.\end{aligned}$$

Durch mehrfaches Anwenden des Distributiv- und Assoziativgesetzes folgt daraus:

$$\vec{q} = \underbrace{(x_1\vec{b}_1 + x_2\vec{b}_2 + \cdots + x_n\vec{b}_n)}_{\text{Vektor } \vec{v}} + \underbrace{(s_1\vec{b}_1 + s_2\vec{b}_2 + \cdots + s_n\vec{b}_n)}_{\text{Vektor } \vec{\alpha}}\,.$$

Da $x_1, x_2, \ldots, x_n \in \mathbb{Z}$ sind, folgt daraus, dass Vektor $\vec{v} \in \mathcal{L}(B)$ ist. Analog: Da $s_1, s_2, \ldots, s_n \in [0, 1)$ sind, folgt daraus, dass Vektor $\vec{\alpha} \in \mathcal{P}(B)$ ist. Eine solche Zerlegung existiert.

Als Letztes muss noch gezeigt werden, dass diese Zerlegung eindeutig ist.

Annahme: Es existieren zwei Vektoren $\vec{v}, \vec{v}' \in \mathcal{L}(B)$ und zwei Vektoren $\vec{\alpha}, \vec{\alpha}' \in \mathcal{P}(B)$, sodass $\vec{q} = \vec{v} + \vec{\alpha} = \vec{v}' + \vec{\alpha}'$ ist. Es gilt:

$$\begin{aligned}\vec{q} &= (x_1 + s_1)\vec{b}_1 + (x_2 + s_2)\vec{b}_2 + \cdots + (x_n + s_n)\vec{b}_n \\ &= (x'_1 + s'_1)\vec{b}_1 + (x'_2 + s'_2)\vec{b}_2 + \cdots + (x'_n + s'_n)\vec{b}_n\,.\end{aligned}$$

Da die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ linear unabhängig sind, muss, damit die Gleichung erfüllt ist,

$$x_i + s_i = x'_i + s'_i, \text{ für } i \in \{1, 2, \ldots, n\}$$

gelten. Durch Umstellen ergibt sich:

$$x_i - x'_i = s'_i - s_i\,.$$

Da $x_i - x'_i \in \mathbb{Z}$ ist, folgt daraus, dass auch $s'_i - s_i \in \mathbb{Z}$ ist. Da sowohl $s_i \in [0, 1)$, als auch $s'_i \in [0, 1)$ ist, muss $s'_i - s_i = 0$ sein und somit gilt $s'_i = s_i$. Somit sind die beiden Vektoren $\vec{\alpha}, \vec{\alpha}' \in \mathcal{P}(B)$ identisch. Aus der Gleichung $\vec{v} + \vec{\alpha} = \vec{v}' + \vec{\alpha}'$ folgt, dass $\vec{v} = \vec{v}'$ gelten muss. Annahme war falsch. Die Zerlegung $\vec{q} = \vec{v} + \vec{\alpha}$ ist eindeutig.

□

Folgerung 2.22 *Jeder Gittervektor $\vec{w} \in \mathcal{L}(B)$ kann in zwei Vektoren $\vec{v} \in \mathcal{L}(B)$ und $\vec{\alpha} \in \mathcal{P}(B)$ zerlegt werden, sodass $\vec{v} + \vec{\alpha}$ mit $\vec{\alpha} = \vec{0}$ gilt.*

Folgerung 2.23 *Jedes Gitter $\mathcal{L}(B)$ zusammen mit der Grundmasche $\mathcal{P}(B)$ erzeugt die lineare Hülle* $\mathrm{span}(B)$ *des Gitters.*

Anmerkung: Mithilfe von Satz 2.21 kann eine surjektive Abbildung eingeführt werden, die alle Vektoren in der linearen Hülle $\mathrm{span}(B)$ auf Vektoren innerhalb der Grundmasche eines Gitters $\mathcal{L}(B)$ abbildet. Dazu wird ein beliebiger Vektor $\vec{q} \in \mathrm{span}(B)$ in zwei Vektoren $\vec{v}$ und $\vec{\alpha}$ zerlegt, sodass $\vec{q} = \vec{v} + \vec{\alpha}$ gilt, wobei $\vec{v}$ ein Gittervektor und $\vec{\alpha}$ ein Vektor innerhalb der Grundmasche ist. Der Vektor $\vec{q}$ kann dann durch Subtraktion des Gittervektors $\vec{v}$ auf den Vektor $\vec{\alpha}$ innerhalb der Grundmasche des Gitters abgebildet werden. Jeder Gittervektor wird bei einer solchen Abbildung auf den Ursprung des Gitters reduziert, da in diesem Fall $\vec{\alpha} = \vec{0}$ gilt.

Die Grundmasche kann bspw. benutzt werden, um zu überprüfen, ob zwei Basen $B, B' \in \mathbb{R}^{m \times n}$ mit $m \geq n$ ein äquivalentes Gitter erzeugen.

Satz 2.24 *Die Matrix $B' \in (\mathcal{L}(B) \backslash \{\vec{0}\})$ ist keine Basis für das Gitter $\mathcal{L}(B)$ genau dann, wenn $\mathcal{P}(B')$ neben dem Ursprung mindestens einen weiteren Gittervektor enthält.*

Beweis:

„$\Rightarrow$“: Da $B, B' \in (\mathcal{L}(B) \backslash \{\vec{0}\})$ ist, muss eine Matrix $M \in \mathbb{Z}^{n \times n}$ existieren, sodass $B' = B \cdot M$ gilt. Da $B' \in \mathcal{L}(B)$ ist, muss $|\det(M)| \neq 0$ sein. Da B' weiter keine Basis für das Gitter $\mathcal{L}(B)$ ist, so muss nach Satz 2.12 $|\det(M)| > 1$ gelten. Daraus folgt, dass $M^{-1} \in \mathbb{R}^{n \times n}$ sein muss. Damit neben dem Ursprung ein weiterer Gittervektor in der Menge $\mathcal{P}(B')$ enthalten ist, muss ein Matrixelement in $M^{-1} \in \mathbb{R}^{n \times n}$ im Intervall $[0, 1)$ liegen. Dies folgt aus der Berechnung der inversen Matrix (Proposition 2.7):

$$M^{-1} = \frac{M_{\mathrm{adj}}}{\det(M)} .$$

Da die Adjunkte M_{adj} der Matrix M mithilfe von Streichungsmatrizen und Unterdeterminanten berechnet wird, muss mindestens ein Element in der Matrix M_{adj} kleiner als der Absolutwert der Determinante von M sein. Da $|\det(M)| > 1$ ist, folgt daraus, dass mindestens ein Matrixelement in der inversen Matrix M^{-1} im Intervall $[0, 1)$ liegen muss. Daraus folgt, dass neben dem Ursprung ein weiterer Gittervektor in der Menge $\mathcal{P}(B')$ enthalten sein muss.

Als Hilfestellung für den letzten Teil des Beweises wird ein Beispiel zur Berechnung einer inversen Matrix anhand der Adjunkten einer Matrix angegeben. Dieses Beispiel benutzt

ausnahmsweise für $M \in \mathbb{Z}^{n\times n}$ eine 3×3-Matrix, da hier das Prinzip der Streichungsmatrizen und Unterdeterminaten anschaulicher ist. Gegeben sei die folgende Matrix:

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

Die Adjunkte der Matrix M ergibt:

$$M_{\text{adj}} = \begin{pmatrix} \det\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} & \det\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} & \det\begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} \\ \det\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} & \det\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \det\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ \det\begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} & \det\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \det\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} .$$

Da $\det(M) = 2$ ist, folgt daraus, dass die inverse Matrix M^{-1} gegeben ist durch:

$$M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

„$\Leftarrow$“: Wenn $\mathcal{P}(B')$ neben dem Ursprung einen weiteren beliebigen Gittervektor $\vec{q} \in \mathcal{L}(B)$ enthält, so muss dieser durch eine reelle Linearkombination der Basisvektoren mit Koeffizienten im Intervall $[0, 1)$ beschreibbar sein:

$$\vec{q} = s_1\vec{b}_1 + s_2\vec{b}_2 + \cdots + s_n\vec{b}_n \text{ , wobei } s_1, s_2, \ldots, s_n \in [0, 1) \text{ ist.}$$

Dies widerspricht der Definition der Gittervektoren, die nur durch ganzzahlige Linearkombinationen beschrieben werden können. Somit müssen die Basen B und B' jeweils andere Gitter erzeugen.

□

Die Grundmasche ist im Gegensatz zu den erzeugten Gittervektoren abhängig von der gewählten Basis. Die Abbildung 2.5 zeigt ein zweidimensionales Gitter ($m = n = 2$) mit zwei unterschiedlichen Gitterbasen und den zugehörigen Grundmaschen. Dabei bestehen die Spalten der Matrix B aus den Gittervektoren $\vec{b}_1, \vec{b}_2$ und die Spalten der Matrix B' entsprechend aus den Gittervektoren $\vec{b}'_1, \vec{b}'_2$.

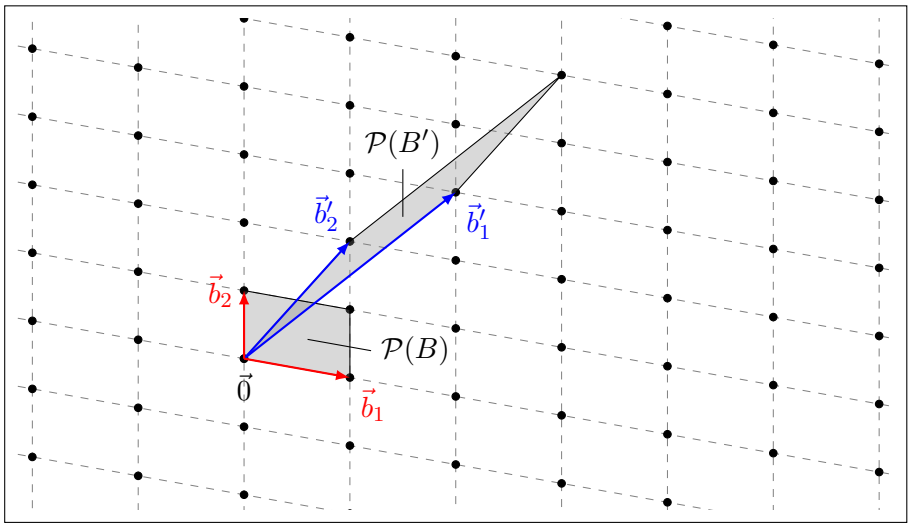


Abbildung 2.5: Ein zweidimensionales Gitter, zwei unterschliedliche Gitterbasen und die zugehörigen Grundmaschen.

Definition 2.25 *Das n-dimensionale Volumen der Grundmasche $\mathcal{P}(B)$ des Gitters $\mathcal{L}(B) \subset \mathbb{R}^m$ wird als* Determinante des Gitters $\mathcal{L}$ *bezeichnet und wie folgt berechnet:*

$$\det(\mathcal{L}(B)) =_{\text{def}} \text{vol}(\mathcal{P}(B)) = \sqrt{|\det(B^T \cdot B)|}\,.$$

Die Berechnung des Volumens der Grundmasche $\mathcal{P}(B)$ wird mithilfe der Gram-Matrix $B^T \cdot B$ berechnet, deren Determinante dem Quadrat des n-dimensionalen Volumens des durch die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ aufgespannten Spates entspricht. In einem eindimensionalen Gitter heißt dieses Volumen auch Länge und in einem zweidimensionalen Gitter entsprechend Fläche.

Hilfssatz 2.26 *In einem vollständigen Gitter $\mathcal{L}(B) \subset \mathbb{R}^m$ mit $m = n$ gilt:*

$$\det(\mathcal{L}(B)) = |\det(B)|\,.$$

Beweis:

Da $\mathcal{L}(B)$ ein vollständiges Gitter ist und somit $m = n$ gilt, ist jede Basismatrix eine quadratische $n \times n$-Matrix. Aus diesem Grund gilt: $\det(B^T) = \det(B)$. Daraus folgt:

$$\begin{aligned}\det(\mathcal{L}(B)) &= \sqrt{|\det(B^T \cdot B)|} = \sqrt{|\det(B^T) \cdot \det(B)|} = \sqrt{|\det(B) \cdot \det(B)|} \\ &= \sqrt{|\det(B)| \cdot |\det(B)|} = \sqrt{|\det(B)|^2} = |\det(B)|\,.\end{aligned}$$

□

Satz 2.27 *Seien $B, B' \in \mathbb{R}^{m\times n}$ mit $m \geq n$ zwei beliebige Basen für das Gitter $\mathcal{L} \subset \mathbb{R}^m$. Dann gilt:*

$$\det(\mathcal{L}(B')) = \det(\mathcal{L}(B))\,.$$

Das bedeutet, dass die Determinante des Gitters $\mathcal{L}$ und somit auch das n-dimensionale Volumen der Grundmasche unabhängig von der gewählten Gitterbasis ist.

Beweis:

Da $B, B' \in \mathbb{R}^{m\times n}$ mit $m \geq n$ Basen des Gitters $\mathcal{L}$ sind, gilt nach Satz 2.12:

$$B' = B \cdot U \text{ mit } U \in \mathbb{Z}^{n\times n} \text{ und } |\det(U)| = 1\,.$$

Somit folgt für die Determinante des Gitters $\mathcal{L}(B')$:

$$\det(\mathcal{L}(B')) = \sqrt{|\det(B'^T \cdot B')|} = \sqrt{|\det((B\cdot U)^T \cdot (B\cdot U)|} = \sqrt{|\det(U^T \cdot B^T \cdot B \cdot U)|}\,.$$

Da die Matrizen U^T, U und das Matrixprodukt $B^T \cdot B$ quadratische Matrizen sind, gilt für die Determinante nach Proposition 2.6:

$$= \sqrt{|\det(U^T)\cdot\det(B^T\cdot B)\cdot\det(U)|} = \sqrt{|\det(U^T)\cdot\det(U)\cdot\det(B^T\cdot B)|}\,.$$

Da die Matrix U unimodular ist, somit auch die transponierte Matrix U^T unimodular ist und die beiden Matrizen quadratisch sind, gilt $\det(U^T) = \det(U)$. Daraus folgt, dass $\det(U^T)\cdot\det(U) = \det(U)\cdot\det(U) = \det(U)^2 = 1$ ist, da $\det(U)$ in der Menge $\{-1, 1\}$ enthalten ist. Daraus folgt weiter:

$$= \sqrt{|1\cdot\det(B^T\cdot B)|} = \sqrt{|\det(B^T\cdot B)|} = \det(\mathcal{L}(B))\,.$$

Somit folgt die Gleichheit der beiden Gitterdeterminanten: $\det(\mathcal{L}(B')) = \det(\mathcal{L}(B))\,.$

□

Mithilfe der Gitterdeterminanten können Abschätzungen über die minimale Anzahl der Gittervektoren in Teilmengen der linearen Hülle abgeleitet werden.

Satz 2.28 *(Satz von Blichfeldt)*

Für jedes $\mathcal{L}(B) \subset \mathbb{R}^m$ und jede messbare Teilmenge $S \subseteq \text{span}(B)$ gilt: Wenn S das n-dimensionale Volumen $\text{vol}(S) > \det(\mathcal{L})$ besitzt, so müssen mindestens zwei unterschiedliche Vektoren $\vec{q}_1, \vec{q}_2 \in S$ existieren, sodass $\vec{q}_1 - \vec{q}_2 \in (\mathcal{L}\backslash\{\vec{0}\})$ ist. Die Differenz der beiden Vektoren $\vec{q}_1, \vec{q}_2 \in S$ ist somit ein nichtverschwindender Vektor im Gitter $\mathcal{L}$.

Beweis: (nach [MG02, S. 11 f.])

Hinweis: Da dieser Beweis etwas komplexer ist, werden die Ideen des Beweises anhand von Abbildungen eines konkreten zweidimensionalen Gitters ($m = n = 2$) veranschaulicht.

Sei S eine Teilmenge der linearen Hülle $\text{span}(B)$ und $\vec{q}_1, \vec{q}_2 \in S$ zwei beliebige Vektoren in dieser Teilmenge. Dann gilt nach Satz 2.21, dass $\vec{q}_1$ in zwei Vektoren $\vec{v} \in \mathcal{L}$ und $\vec{\alpha} \in \mathcal{P}(B)$ zerlegt werden kann, sodass gilt:

$$\vec{q}_1 = \vec{v} + \vec{\alpha}\,.$$

Analog kann $\vec{q}_2$ in zwei Vektoren $\vec{w} \in \mathcal{L}$ und $\vec{\beta} \in \mathcal{P}(B)$ zerlegt werden, sodass gilt:

$$\vec{q}_2 = \vec{w} + \vec{\beta}\,.$$

Für jeden Gittervektor $\vec{l} \in \mathcal{L}$ wird die Menge $\mathcal{P}_{\vec{l}}(B)$ wie folgt definiert:

$$\mathcal{P}_{\vec{l}}(B) =_{\text{def}} \{\vec{l} + \vec{\gamma} : \vec{\gamma} \in \mathcal{P}(B)\}\,.$$

Die Menge $\mathcal{P}_{\vec{l}}(B)$ beinhaltet alle Vektoren der Grundmasche $\mathcal{P}(B)$, die um einen bestimmten Stützvektor $\vec{l} \in \mathcal{L}$ verschoben wurden.

Die Abbildung 2.6 zeigt die Ausgangssituation des Beweises.

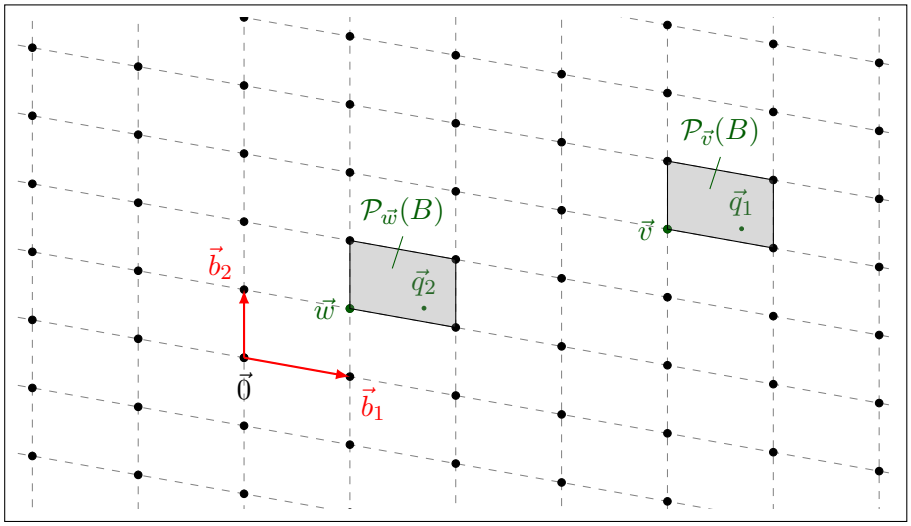


Abbildung 2.6: Zwei beliebige Vektoren $\vec{q}_1, \vec{q}_2 \in \mathbb{R}^2$ im zweidimensionalen Raum.

Die Menge S wird für jeden Gittervektor $\vec{l} \in \mathcal{L}$ in eine Partition $S_{\vec{l}}$ wie folgt zerlegt:

$$S_{\vec{l}} =_{\text{def}} S \cap (\mathcal{P}_{\vec{l}}(B))\,.$$

Die Abbildung 2.7 zeigt die Partitionierung der Menge S für das Beispiel aus Abbildung 2.6.

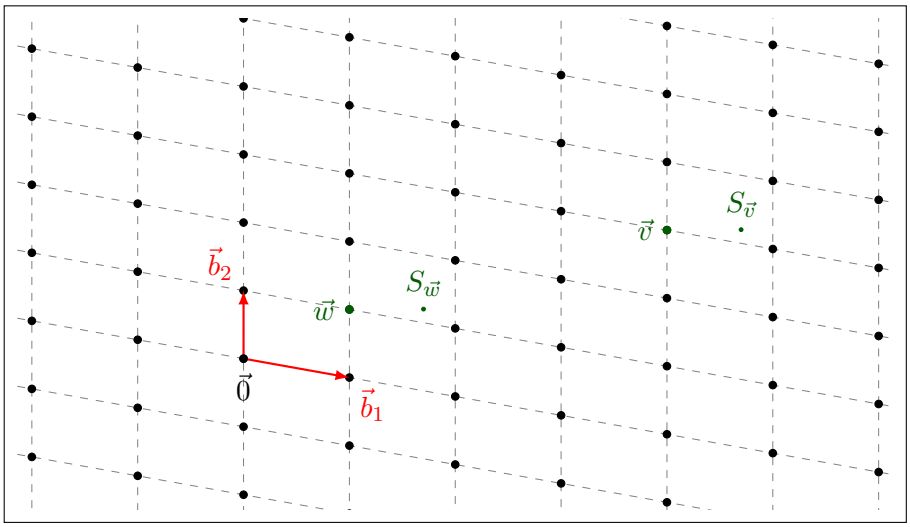


Abbildung 2.7: Ein Beispiel für eine Partition der Menge S.

Alle Partitionen der Menge S sind paarweise disjunkt und somit gilt:

$$S = \bigcup_{\vec{l} \in \mathcal{L}} S_{\vec{l}}.$$

Da $\mathcal{L}$ abzählbar ist, so ist auch die Vereinigungsmenge der Partitionen $S_{\vec{l}}$ abzählbar und es gilt:

$$\mathrm{vol}(S) = \sum_{\vec{l} \in \mathcal{L}} \mathrm{vol}(S_{\vec{l}})\,.$$

Alle Partitionen werden nun so verschoben, dass diese in der Grundmasche $\mathcal{P}(B)$ liegen. Für jeden Gittervektor $\vec{l} \in \mathcal{L}$ wird die Menge $T_{\vec{l}}$ wie folgt definiert:

$$T_{\vec{l}} =_{\mathrm{def}} \{\vec{\gamma} - \vec{l} : \vec{\gamma} \in S_{\vec{l}}\}\,.$$

Die Vereinigungsmenge aller verschobenenen Partitionen $T_{\vec{l}}$ liegt komplett in der Grundmasche $\mathcal{P}(B)$ und das Volumen von S hat sich durch die Verschiebung nicht geändert. Es gilt weiterhin:

$$\mathrm{vol}(S) = \sum_{\vec{l} \in \mathcal{L}} \mathrm{vol}(S_{\vec{l}}) = \sum_{\vec{l} \in \mathcal{L}} \mathrm{vol}(T_{\vec{l}})\,.$$

Die Abbildung 2.8 zeigt die zur Grundmasche $\mathcal{P}(B)$ verschobenen Partitionen der Menge S aus Abbildung 2.7.

Es bleibt zu zeigen, dass die verschobenen Partitionen $T_{\vec{l}}$ nicht paarweise disjunkt sein können.

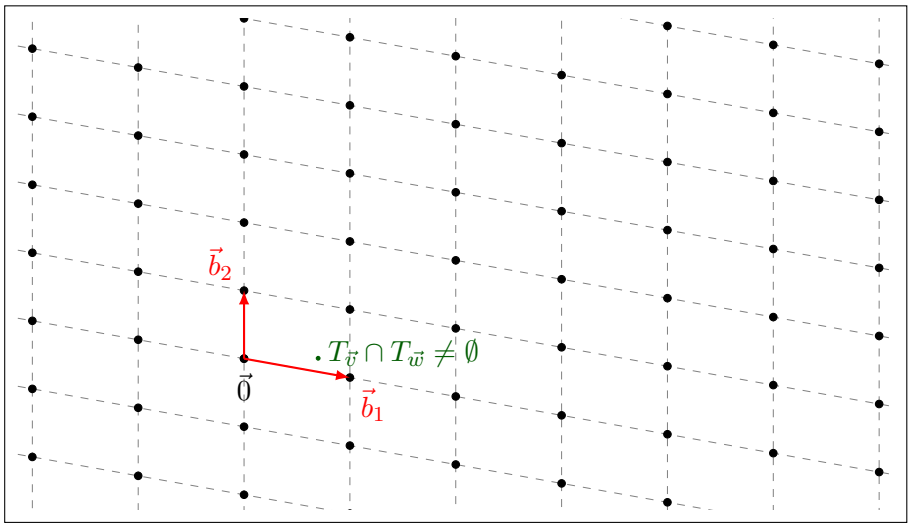


Abbildung 2.8: Die zur Grundmasche verschobenen Partitionen der Menge S.

Annahme: Die Partitionen $T_{\vec{l}}$ sind paarweise disjunkt, dann gilt für das Volumen von S:

$$\text{vol}(S) = \sum_{\vec{l} \in \mathcal{L}} \text{vol}(T_{\vec{l}}) = \text{vol}\left(\bigcup_{\vec{l} \in \mathcal{L}} T_{\vec{l}} \right) \leq \text{vol}(\mathcal{P}(B)).$$

Aus dem zu beweisenden Satz gilt weiter:

$$\det(\mathcal{L}) < \text{vol}(S)\,.$$

Zusammen:

$$\det(\mathcal{L}) < \text{vol}(S) \leq \text{vol}(\mathcal{P}(B))\,.$$

Daraus folgt:

$$\det(\mathcal{L}) < \text{vol}(\mathcal{P}(B))\,.$$

Widerspruch zu Hilfssatz 2.26, da $\det(\mathcal{L}) = \text{vol}(\mathcal{P}(B))$ gilt. Annahme war falsch. Die verschobenen Partitionen $T_{\vec{l}}$ sind nicht paarweise disjunkt.

Es existieren mindestens zwei unterschiedliche Partitionen $T_{\vec{v}}, T_{\vec{w}}$ wobei $\vec{v}, \vec{w} \in \mathcal{L}$, sodass $T_{\vec{v}} \cap T_{\vec{w}} \neq \emptyset$ ist. Sei $\vec{\alpha}$ ein Vektor in der nichtleeren Schnittmenge von $T_{\vec{v}} \cap T_{\vec{w}}$. Dann gilt weiter $\vec{\alpha} \in \mathcal{P}(B)$ und die nicht verschobenen Vektoren $\vec{q}_1 \in S_{\vec{v}}$ und $\vec{q}_2 \in S_{\vec{w}}$ können wie folgt dargestellt werden:

$$\vec{q}_1 = \vec{v} + \vec{\alpha}$$
$$\vec{q}_2 = \vec{w} + \vec{\alpha}\,.$$

Da $\vec{v} \neq \vec{w}$ gilt, gilt auch $S_{\vec{v}} \cap S_{\vec{w}} = \emptyset$ und somit auch $\vec{q}_1 \neq \vec{q}_2$. Die Differenz von $\vec{q}_1$ und $\vec{q}_2$

ergibt:

$$\vec{q}_1 - \vec{q}_2 = \vec{v} + \vec{\alpha} - (\vec{w} + \vec{\alpha}) = \vec{v} + \vec{\alpha} - \vec{w} - \vec{\alpha} = \vec{v} - \vec{w} \in \mathcal{L}\,.$$

Die Differenz der beiden Vektoren $\vec{q}_1$ und $\vec{q}_2$ ist somit ein nichtverschwindender Gittervektor des Gitters $\mathcal{L}$. Die Abbildung 2.9 zeigt das Resultat des Beweises.

□

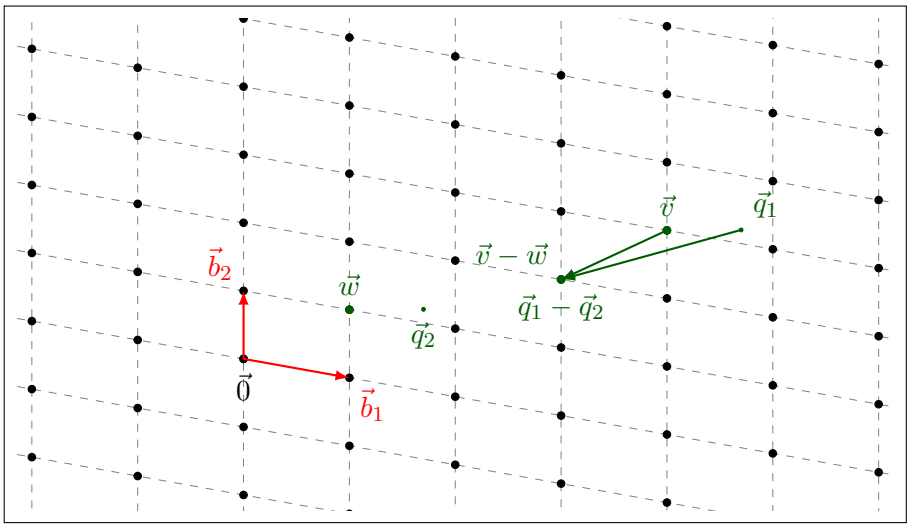


Abbildung 2.9: Die Differenz der beiden Vektoren $\vec{q}_1, \vec{q}_2 \in \mathbb{R}^2$ ist ein Gittervektor.

Folgerung 2.29 *(Satz von Minkowski)*

Für jedes Gitter $\mathcal{L}(B)$ mit dem Rang n und jede beliebige konvexe Menge $S \subset \mathrm{span}(B)$, die symmetrisch um den Ursprung des Gitters $\mathcal{L}$ ist, gilt: Wenn $\mathrm{vol}(S) > 2^n \det(\mathcal{L})$ ist, dann enthält S mindestens einen nichtverschwindenden Gittervektor $\vec{v} \in (S \cap \mathcal{L} \backslash \{\vec{0}\})$.

Beweis: (nach [MG02, S. 12])

Sei die Teilmenge S' der Menge S gegeben durch $S' =_{\mathrm{def}} \{\vec{q} : 2\vec{q} \in S\}$

Für das Volumen der Menge S gilt:

$$\mathrm{vol}(S) > 2^n \det(\mathcal{L})\,.$$

Da n der Rang des Gitters $\mathcal{L}$ ist und somit $n \geq 1$ und damit auch $2^n > 0$ gilt, gilt für das Volumen von S':

$$\frac{1}{2^n}\,\mathrm{vol}(S) = \mathrm{vol}(S') > \det(\mathcal{L})\,.$$

Aus diesem Grund existieren nach Satz 2.28 zwei unterschiedliche Vektoren $\vec{q}_1, \vec{q}_2 \in S'$, sodass die Differenz der beiden Vektoren $\vec{q}_1 - \vec{q}_2 \in (\mathcal{L} \backslash \{\vec{0}\})$ ist.

Da $S' =_{\text{def}} \{\vec{q} : 2\vec{q} \in S\}$ ist, sind die Vektoren $2\vec{q}_1$ und $2\vec{q}_2$ in der Menge S enthalten. Da S zusätzlich symmetrisch zum Ursprung des Gitters $\mathcal{L}$ ist, gilt auch $-2\vec{q}_1, -2\vec{q}_2 \in S$. Des Weiteren ist S konvex und somit gilt weiterhin, dass der Mittelpunkt der Strecke von $-2\vec{q}_2$ nach $2\vec{q}_1$ auch in S enthalten ist. Dieser Mittelpunkt liegt bei

$$\frac{1}{2} \cdot (2\vec{q}_1 + (-2\vec{q}_2)) = \frac{2\vec{q}_1 + (-2\vec{q}_2)}{2} = \frac{2\vec{q}_1 - 2\vec{q}_2}{2} = \frac{2(\vec{q}_1 - \vec{q}_2)}{2} = \vec{q}_1 - \vec{q}_2 \in S\,.$$

Damit ist $\vec{v} = \vec{q}_1 - \vec{q}_2$ sowohl ein nichtverschwindender Gittervektor, als auch in der Menge S enthalten. Dies zeigt, das die Menge S mindestens einen nichtverschwindenden Gittervektor enthalten muss. Die Abbildung 2.10 verdeutlicht die einzelnen Schritte des Beweises.

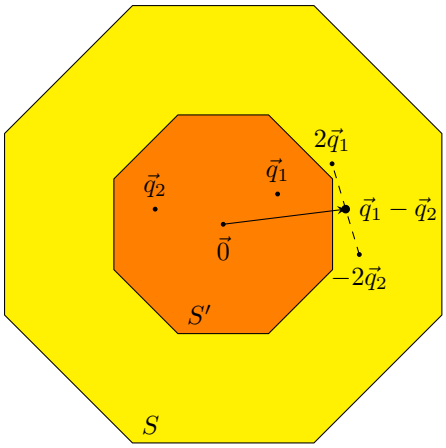


Abbildung 2.10: Die Menge S muss mindestens einen nichtverschwindenden Gittervektor enthalten.

□

In Folgerung 2.29 wird vorausgesetzt, dass die Menge S zwei Eigenschaften erfüllt: Sie muss symmetrisch zum Ursprung des Gitters und konvex sein. Die Abbildung 2.11 zeigt, dass eine dieser beiden Eigenschaften nicht ausreicht bzw. dann Gegenbeispiele existieren, sodass $\text{vol}(S) > 2^n \det(\mathcal{L})$ gilt, die Menge S aber trotzdem keinen Gittervektor enthält.

2.3 Gram-Schmidt Orthogonalisierung

Die Basis eines Gitters besteht aus n linear unabhängigen Vektoren. Diese linear unabhängigen Vektoren können noch in eine sogenannte Orthogonalbasis transformiert werden, sodass alle Basisvektoren paarweise orthogonal sind. Aus einer solchen Orthogonalbasis können weitere Eigenschaften eines Gitters abgeleitet werden.

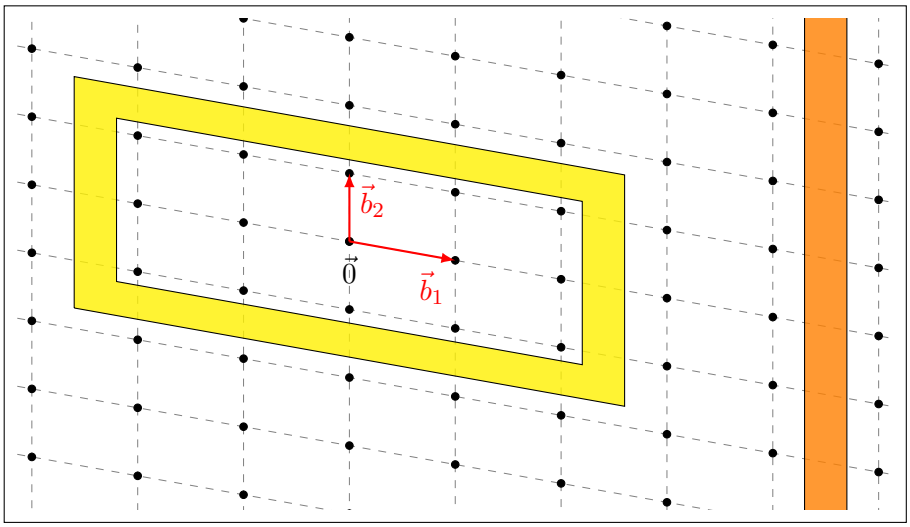


Abbildung 2.11: Linke Fläche: Symmetrisch, nicht konvex. Rechte Fläche: Konvex, nicht symmetrisch.

Anmerkung: Im Folgenden wird das Skalarprodukt von zwei Vektoren mit $\langle \cdot, \cdot \rangle$ symbolisiert.

Satz 2.30 *(Gram-Schmidt Orthogonalisierung)*

Seien $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{R}^m$ *Basisvektoren des Gitters* $\mathcal{L}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$*, dann können diese durch folgendes Verfahren*

$$\vec{o}_i = \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)\,, \text{ wobei } p_{\vec{o}_j}(\vec{b}_i) = \frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} \vec{o}_j \text{ ist und für } i \in \{1, 2, \ldots, n\},$$

in eine Orthogonalbasis $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n$ *transformiert werden. Das Verfahren benutzt* $p_{\vec{o}_j}(\vec{b}_i)$*, um den Vektor* $\vec{b}_i$ *auf den Vektor* $\vec{o}_j$ *zu projizieren. Da* $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n$ *eine Orthogonalbasis ist, sind alle Vektoren dieser Basis paarweise orthogonal.*

Beweis:

Es wird nicht bewiesen, dass für jeden Vektorraum, der span(B) abdeckt, eine Orthogonalbasis existieren muss. Ein Beweis für diese Aussage befindet sich bspw. in [Klo97, S. 192].

Der Satz 2.30 wird mithilfe einer Induktion gezeigt:

(IA)

Nach dem zu beweisenden Satz gilt für $i = 1$:

$$\vec{o}_1 = \vec{b}_1\,.$$

Da für $i = 1$ nur ein Vektor $\vec{o}_1$ existiert, ist dieser orthogonal.

Aufgrund des sehr einfachen Beispiels von $i = 1$ wird hier ein weiteres Beispiel für den Induktionsanfang gegeben werden. Nach dem zu beweisenden Satz gilt für $i = 2$:

$$\begin{aligned}\vec{o}_1 &= \vec{b}_1\\ \vec{o}_2 &= \vec{b}_2 - p_{\vec{o}_1}(\vec{b}_2)\,.\end{aligned}$$

Es bleibt zu zeigen, dass die beiden Vektoren $\vec{o}_1$ und $\vec{o}_2$ orthogonal zueinander sind, also $\langle \vec{o}_1, \vec{o}_2\rangle = \langle \vec{o}_2, \vec{o}_1\rangle = 0$ gültig ist. Durch Einsetzen der Formel aus dem Gram-Schmidt Orthogonalisierungsverfahren ergibt sich für das Skalarprodukt $\langle \vec{o}_2, \vec{o}_1\rangle$:

$$\langle \vec{o}_2, \vec{o}_1\rangle = \langle \vec{b}_2 - p_{\vec{o}_1}(\vec{b}_2), \vec{o}_1\rangle\,.$$

Aufgrund der Bilinearität des Skalarprodukts in $\mathbb{R}^m$ gilt weiter:

$$\begin{aligned}\langle \vec{b}_2 - p_{\vec{o}_1}(\vec{b}_2), \vec{o}_1\rangle &= \langle \vec{b}_2, \vec{o}_1\rangle - \langle p_{\vec{o}_1}(\vec{b}_2), \vec{o}_1\rangle\\ &= \langle \vec{b}_2, \vec{o}_1\rangle - \left\langle \frac{\langle \vec{b}_2, \vec{o}_1\rangle}{\langle \vec{o}_1, \vec{o}_1\rangle}\vec{o}_1, \vec{o}_1 \right\rangle\\ &= \langle \vec{b}_2, \vec{o}_1\rangle - \frac{\langle \vec{b}_2, \vec{o}_1\rangle}{\langle \vec{o}_1, \vec{o}_1\rangle}\langle \vec{o}_1, \vec{o}_1\rangle\\ &= \langle \vec{b}_2, \vec{o}_1\rangle - \langle \vec{b}_2, \vec{o}_1\rangle\\ &= 0\,.\end{aligned}$$

Die beiden Vektoren $\vec{o}_1$ und $\vec{o}_2$ sind orthogonal zueinander.

(IV)

Die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{i-1} \in \mathbb{R}^m$ eines Vektorraums $\text{span}(B)$ können durch folgendes Verfahren

$$\vec{o}_k = \vec{b}_k - \sum_{j=1}^{k-1} p_{\vec{o}_j}(\vec{b_k})\,,\ \text{wobei}\ p_{\vec{o}_j}(\vec{b_k}) = \frac{\langle \vec{b_k}, \vec{o}_j\rangle}{\langle \vec{o}_j, \vec{o}_j\rangle}\vec{o}_j\ \text{ist und für}\ k \in \{1, 2, \ldots, (i-1)\},$$

in Vektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{i-1} \in \mathbb{R}^m$ transformiert werden, sodass die Vektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{i-1}$ paarweise orthogonal sind.

(IS)

$(i-1) \to i$:

Nach Induktionsvoraussetzung sind die Vektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{i-1}$ paarweise orthogonal und der Vektor $\vec{o}_i$ ist gegeben durch:

$$\vec{o}_i = \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)\,,\ \text{wobei}\ p_{\vec{o}_j}(\vec{b}_i) = \frac{\langle \vec{b}_i, \vec{o}_j\rangle}{\langle \vec{o}_j, \vec{o}_j\rangle}\vec{o}_j\ \text{ist.}$$

Es muss gezeigt werden, dass der erzeugte Vektor $\vec{o}_i$ paarweise orthogonal zu allen Vektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{i-1}$ ist. Also, dass $\langle \vec{o}_k, \vec{o}_i \rangle = \langle \vec{o}_i, \vec{o}_k \rangle = 0$ für alle $k \in \{1, 2, \ldots, (i-1)\}$ gültig ist. Durch Einsetzen der Formel aus dem Gram-Schmidt Orthogonalisierungsverfahren ergibt sich für das Skalarprodukt $\langle \vec{o}_i, \vec{o}_k \rangle$:

$$\langle \vec{o}_i, \vec{o}_k \rangle = \left\langle \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_k \right\rangle .$$

Aufgrund der Bilinearität des Skalarprodukts in $\mathbb{R}^m$ gilt:

$$\left\langle \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_k \right\rangle = \langle \vec{b}_i, \vec{o}_k \rangle - \left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_k \right\rangle .$$

Die Summe $\sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)$ wird im Folgenden durch eine alternative Schreibweise ersetzt:

$$\sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) = (p_{\vec{o}_1}(\vec{b}_i) + p_{\vec{o}_2}(\vec{b}_i) + \cdots + p_{\vec{o}_{i-1}}(\vec{b}_i)) .$$

Dies ergibt weiter:

$$\langle \vec{b}_i, \vec{o}_k \rangle - \left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_k \right\rangle = \langle \vec{b}_i, \vec{o}_k \rangle - \langle (p_{\vec{o}_1}(\vec{b}_i) + p_{\vec{o}_2}(\vec{b}_i) + \cdots + p_{\vec{o}_{i-1}}(\vec{b}_i)), \vec{o}_k \rangle .$$

Das Skalarprodukt $\langle (p_{\vec{o}_1}(\vec{b}_i) + p_{\vec{o}_2}(\vec{b}_i) + \cdots + p_{\vec{o}_{i-1}}(\vec{b}_i)), \vec{o}_k \rangle$ kann wiederum aufgrund der Bilinearität des Skalarprodukts in $\mathbb{R}^m$ in mehrere Skalarprodukte zerlegt werden:

$$= \langle \vec{b}_i, \vec{o}_k \rangle - (\langle p_{\vec{o}_1}(\vec{b}_i), \vec{o}_k \rangle + \langle p_{\vec{o}_2}(\vec{b}_i), \vec{o}_k \rangle + \cdots + \langle p_{\vec{o}_{i-1}}(\vec{b}_i)), \vec{o}_k \rangle) .$$

Alle Skalarprodukte $\langle p_{\vec{o}_j}(\vec{b}_i)), \vec{o}_k \rangle$ für $j, k \in \{1, 2, \ldots, (i-1)\}$ und $j \neq k$ ergeben 0, da die Vektoren $\vec{o}_j$ und $\vec{o}_k$ nach Induktionsvoraussetzung paarweise orthogonal zueinander sind. Somit ist auch der auf $\vec{o}_j$ projizierte Vektor $p_{\vec{o}_j}(\vec{b}_i)$ orthogonal zu $\vec{o}_k$. Das einzige von Null verschiedene Skalarprodukt aus der Summenformel ist: $\langle p_{\vec{o}_k}(\vec{b}_i), \vec{o}_k \rangle$.

Daraus folgt:

$$\begin{aligned}
\langle \vec{o}_i, \vec{o}_k \rangle &= \langle \vec{b}_i, \vec{o}_k \rangle - \langle p_{\vec{o}_k}(\vec{b}_i), \vec{o}_k \rangle \\
&= \langle \vec{b}_i, \vec{o}_k \rangle - \left\langle \frac{\langle \vec{b}_i, \vec{o}_k \rangle}{\langle \vec{o}_k, \vec{o}_k \rangle} \vec{o}_k, \vec{o}_k \right\rangle \\
&= \langle \vec{b}_i, \vec{o}_k \rangle - \frac{\langle \vec{b}_i, \vec{o}_k \rangle}{\langle \vec{o}_k, \vec{o}_k \rangle} \langle \vec{o}_k, \vec{o}_k \rangle \\
&= \langle \vec{b}_i, \vec{o}_k \rangle - \langle \vec{b}_i, \vec{o}_k \rangle \\
&= 0 .
\end{aligned}$$

Dies zeigt, dass alle erzeugten Basisvektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{R}^m$ der Gram-Schmidt Orthogonalisierung paarweise orthogonal sind.

□

Im Folgenden wird für die Gram-Schmidt orthogonalisierten Basisvektoren eines Gitters eine Matrix O angegeben. Dabei besteht die Matrix O bzw. die Gram-Schmidt orthogonalisiere Basis aus den Spaltenvektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n$.

Anmerkung: Eine durch die Gram-Schmidt Orthogonalisierung erzeugte Orthogonalbasis O einer Gitterbasis B besteht nicht zwangsläufig nur aus Gittervektoren. Es kann nach Durchführung der Gram-Schmidt Orthogonalisierung mindestens ein Vektor der Orthogonalbasis O existieren, der nicht im Gitter enthalten ist.

Des Weiteren zeigt die Berechnungsformel der Gram-Schmidt Orthogonalisierung, dass das Verfahren abhängig von der Reihenfolge der Basisvektoren der Gitterbasis B ist. Die Basis $B =_{\text{def}} \left[\vec{b}_1 | \vec{b}_2 | \cdots | \vec{b}_n\right]$ erzeugt somit eine unterschiedliche Gram-Schmidt Orthogonalbasis als die Basis $B' =_{\text{def}} \left[\vec{b}_2 | \vec{b}_1 | \cdots | \vec{b}_n\right]$, obwohl beide Basen ein äquivalentes Gitter erzeugen.

Im Folgenden bezeichnet $\| \cdot \|$ die euklidische Norm eines Vektors.

Satz 2.31 *Sei $B \in \mathbb{R}^{m \times n}$ mit $m \geq n$ die Basis eines Gitters $\mathcal{L}$ und $O \in \mathbb{R}^{m \times n}$ die zugehörige Gram-Schmidt Orthogonalbasis. Sei weiter $\vec{o}_i$ der i. Gram-Schmidt orthogonalisierte Basisvektor und $\vec{b}_i$ der i. Basisvektor des Gitters $\mathcal{L}$ mit $i \in \{1, 2, \ldots, n\}$, so gilt für die Länge dieser Vektoren folgender Zusammenhang:*

$$\|\vec{o}_i\| \leq \|\vec{b}_i\| .$$

Beweis:

Die Berechnungsformel des Gram-Schmidt Orthogonalisierungsverfahren kann wie folgt umgeschrieben werden:

$$\vec{o}_i = \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)$$

$$\vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) = \vec{b}_i .$$

Für die Länge des i. Gram-Schmidt orthogonalisierten Basisvektors gilt:

$$\begin{aligned}
\left\| \vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\| &= \|\vec{b}_i\| \\
\sqrt{\left\langle \vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\rangle} &= \|\vec{b}_i\| \\
\sqrt{\langle \vec{o}_i, \vec{o}_i \rangle + 2 \left\langle \vec{o}_i, \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\rangle + \left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\rangle} &= \|\vec{b}_i\| .
\end{aligned}$$

Da der Gram-Schmidt orthogonalisierte Basisvektor $\vec{o}_i$ orthogonal zu allen $p_{\vec{o}_j}(\vec{b}_i)$ mit $j < i$ ist, gilt $\left\langle \vec{o}_i, \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\rangle = 0$. Daraus folgt weiter:

$$\sqrt{\langle \vec{o}_i, \vec{o}_i \rangle + \left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\rangle} = \|\vec{b}_i\| .$$

Da für alle $k, l \in \{1, 2, \ldots, n\}$ und $k \neq l$ das Skalarprodukt $\langle p_{\vec{o}_k}(\vec{b}_i), p_{\vec{o}_l}(\vec{b}_i) \rangle = 0$ ist, kann $\left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) \right\rangle$ zu $\sum_{j=1}^{i-1} \langle p_{\vec{o}_j}(\vec{b}_i), p_{\vec{o}_j}(\vec{b}_i) \rangle$ vereinfacht werden:

$$\sqrt{\langle \vec{o}_i, \vec{o}_i \rangle + \sum_{j=1}^{i-1} \left\langle p_{\vec{o}_j}(\vec{b}_i), p_{\vec{o}_j}(\vec{b}_i) \right\rangle} = \|\vec{b}_i\| .$$

Da $\sum_{j=1}^{i-1} \left\langle p_{\vec{o}_j}(\vec{b}_i), p_{\vec{o}_j}(\vec{b}_i) \right\rangle = \sum_{j=1}^{i-1} \|p_{\vec{o}_j}(\vec{b}_i)\|^2 \geq 0$ ist, gilt weiter:

$$\begin{aligned}
\sqrt{\langle \vec{o}_i, \vec{o}_i \rangle} \leq \sqrt{\langle \vec{o}_i, \vec{o}_i \rangle + \sum_{j=1}^{i-1} \left\langle p_{\vec{o}_j}(\vec{b}_i), p_{\vec{o}_j}(\vec{b}_i) \right\rangle} &= \|\vec{b}_i\| \\
\|\vec{o}_i\| \leq \sqrt{\langle \vec{o}_i, \vec{o}_i \rangle + \sum_{j=1}^{i-1} \left\langle p_{\vec{o}_j}(\vec{b}_i), p_{\vec{o}_j}(\vec{b}_i) \right\rangle} &= \|\vec{b}_i\| \\
\|\vec{o}_i\| &\leq \|\vec{b}_i\| .
\end{aligned}$$

□

Satz 2.32 *Sei $B \in \mathbb{R}^{m \times n}$ mit $m \geq n$ eine Basis des Gitters $\mathcal{L}$ und $O \in \mathbb{R}^{m \times n}$ die zugehörige Gram-Schmidt Orthogonalbasis, die aus der Basis B entstanden ist, dann gilt:*

$$\det(\mathcal{L}(B)) = \prod_{i=1}^{n} \|\vec{o}_i\| .$$

Beweis:

Die Berechnungsformel des Gram-Schmidt Orthogonalisierungsverfahren kann wie folgt umgeschrieben werden:

$$\vec{o_i} = \vec{b_i} - \sum_{j=1}^{i-1} p_{\vec{o_j}}(\vec{b_i})$$

$$\vec{o_i} + \sum_{j=1}^{i-1} p_{\vec{o_j}}(\vec{b_i}) = \vec{b_i}$$

$$\vec{b_i} = \vec{o_i} + \sum_{j=1}^{i-1} p_{\vec{o_j}}(\vec{b_i})\,.$$

Der Zusammenhang zwischen den Vektoren der Gram-Schmidt Orthogonalbasis O und den Vektoren der Gitterbasis B kann auch als Matrixmultiplikation dargestellt werden:

$$B = O \cdot M\,.$$

Die Basis O besitzt die Basisvektoren der Gram-Schmidt Orthogonalbasis als Spaltenvektoren und die Matrix $M \in \mathbb{R}^{n\times n}$ ist gegeben durch:

$$M =_{\text{def}} \begin{pmatrix} 1 & \frac{\langle \vec{b}_2, \vec{o}_1\rangle}{\langle \vec{o}_1, \vec{o}_1\rangle} & \cdots & \frac{\langle \vec{b}_n, \vec{o}_1\rangle}{\langle \vec{o}_1, \vec{o}_1\rangle} \\ 0 & 1 & \cdots & \frac{\langle \vec{b}_n, \vec{o}_2\rangle}{\langle \vec{o}_2, \vec{o}_2\rangle} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}\,.$$

Es wird kurz hervorgehoben, dass die Matrx M unimodular ist, d. h., dass für die Determinante dieser Matrix $|\det(M)| = 1$ gilt.

Im Folgenden bezeichnet $b_{i,j}$ bzw. $o_{i,j}$ einer Matrix B bzw. O den Eintrag in der entsprechenden Matrix in der i. Zeile und der j. Spalte.

Aus der Multiplikation der Matrix O mit der unimodularen Matrix M ergeben sich die Vektoren der Gitterbasis als Spaltenvektoren der Matrix B:

$$\begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m,1} & b_{m,2} & \cdots & b_{m,n} \end{pmatrix} = \begin{pmatrix} o_{1,1} & o_{1,2} & \cdots & o_{1,n} \\ o_{2,1} & o_{2,2} & \cdots & o_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ o_{m,1} & o_{m,2} & \cdots & o_{m,n} \end{pmatrix} \cdot \begin{pmatrix} 1 & \frac{\langle \vec{b}_2, \vec{o}_1\rangle}{\langle \vec{o}_1, \vec{o}_1\rangle} & \cdots & \frac{\langle \vec{b}_n, \vec{o}_1\rangle}{\langle \vec{o}_1, \vec{o}_1\rangle} \\ 0 & 1 & \cdots & \frac{\langle \vec{b}_n, \vec{o}_2\rangle}{\langle \vec{o}_2, \vec{o}_2\rangle} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}\,.$$

Mithilfe von Hilfssatz 2.26 ergibt sich für die Determinante des Gitters:

$$\det(\mathcal{L}(B)) = \sqrt{|\det(B^T \cdot B)|}\,.$$

Da $B = O \cdot M$ gilt, folgt daraus:

$$= \sqrt{|\det((O \cdot M)^T \cdot O \cdot M)|} = \sqrt{|\det(M^T \cdot O^T \cdot O \cdot M)|}\,.$$

Da die Matrizen M^T, M und das Matrixprodukt $O^T \cdot O$ quadratische Matrizen sind, gilt für die Determinante nach Proposition 2.6:

$$= \sqrt{|\det(M^T) \cdot \det(O^T \cdot O) \cdot \det(M)|}\,.$$

Die Determinante der Matrix M und die Deteriminante der Matrix M^T sind 1, daraus folgt weiter:

$$= \sqrt{|1 \cdot \det(O^T \cdot O) \cdot 1|} = \sqrt{|\det(O^T \cdot O)|}\,.$$

Dies zeigt, dass das Volumen des, durch die Gram-Schmidt Orthogonalbasis, aufgespannten Spates der Gitterdeterminanten $\det(\mathcal{L}(B))$ entspricht. Die Bestimmung der Gram-Matrix $O^T \cdot O$ ergibt:

$$O^T \cdot O = \begin{pmatrix} \langle \vec{o}_1, \vec{o}_1 \rangle & \langle \vec{o}_1, \vec{o}_2 \rangle & \cdots & \langle \vec{o}_1, \vec{o}_n \rangle \\ \langle \vec{o}_2, \vec{o}_1 \rangle & \langle \vec{o}_2, \vec{o}_2 \rangle & \cdots & \langle \vec{o}_2, \vec{o}_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{o}_n, \vec{o}_1 \rangle & \langle \vec{o}_n, \vec{o}_2 \rangle & \cdots & \langle \vec{o}_n, \vec{o}_n \rangle \end{pmatrix}.$$

Innerhalb der Gram-Matrix verschwinden alle Skalarprodukte, die nicht auf der Matrixdiagonalen liegen, da alle Vektoren der Gram-Schmidt Orthogonalbasis paarweise orthogonal sind. Daraus folgt:

$$O^T \cdot O = \begin{pmatrix} \|\vec{o}_1\|^2 & 0 & \cdots & 0 \\ 0 & \|\vec{o}_2\|^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \|\vec{o}_n\|^2 \end{pmatrix}.$$

Mithilfe dieser Umformung lässt sich leicht die Determinante der Matrix $O^T \cdot O$ bestimmen:

$$\sqrt{|\det(O^T \cdot O)|} = \sqrt{\left|\prod_{i=1}^{n} \|\vec{o}_i\|^2\right|} = \sqrt{\prod_{i=1}^{n} \|\vec{o}_i\|^2} = \prod_{i=1}^{n} \|\vec{o}_i\|\,.$$

□

Anmerkung: Die durch die Gram-Schmidt Orthogonalbasis aufgespannte Grundmasche wird auch als *Gram-Schmidt Grundmasche* bezeichnet. Satz 2.32 zeigt, dass die durch das Gram-Schmidt Orthogonalisierungsverfahren erzeugte Basis nicht einfach normalisiert werden darf. Aus den Vektoren einer solchen normalisierten Orthogonalbasis wird nicht zwingend das gleiche Volumen aufgespannt.

Definition 2.33 *Sei* $\mathcal{K}_0(r) = \{\vec{v} \in \mathbb{R}^m : \|\vec{v}\| \in (0, r)\}$ *eine m-dimensionale Kugel, deren Mittelpunkt der Ursprung ist, mit dem Radius* r*. Die Konstanten* $\lambda_1, \lambda_2, \ldots, \lambda_n$ *werden die* sukzessiven Minima *des Gitters genannt, wobei* $\lambda_i(\mathcal{L})$ *der Radius* r *der kleinsten Kugel* $\mathcal{K}_0(r)$ *angibt, die* i *linear unabhängige Vektoren des Gitters* $\mathcal{L}$ *enthält. Die geschlossene Kugel* $\{\vec{v} \in \mathbb{R}^m : \|\vec{v}\| \in [0, r]\}$ *wird mit* $\overline{\mathcal{K}}_0(r)$ *bezeichnet.*

Veranschaulicht enthält das sukzessive Minimum λ_i die i kürzesten Vektoren des Gitters, die einen i-dimensionalen Vektorraum aufspannen. Die Vektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_i$ des sukzessiven Minimum $\lambda_i(\mathcal{L})$ bilden nicht zwingend eine Basis für das Gitter $\mathcal{L}$ (siehe bspw. [Sch08, S. 19], [Ngu10, S. 32 f.] oder [MG02, S. 126]).

Damit die Länge von unterschiedlichen Vektoren im Gitter verglichen werden kann, muss eine Norm eingeführt werden. Bei Betrachtungen von Gittern werden häufig die sogenannten p-Normen verwendet. Dabei ist die allgemeine Norm $\ell_p(\vec{v})$ für $p \geq 1$ und einen Vektor $\vec{v} \in \mathbb{R}^m$ gegeben durch:

$$\ell_p(\vec{v}) = \|\vec{v}\|_p = \sqrt[p]{\sum_{i=1}^{m} |v_i|^p}, \text{ wobei } v_i \text{ die } i\text{. Komponente des Vektors } \vec{v} \text{ ist.}$$

Meistens werden nur die drei Normen ℓ_1 (Summennorm), ℓ_2 (Euklidische Norm) und ℓ_∞ (Maximumsnorm) betrachtet, die durch folgende Formeln gegeben sind:

$$\ell_1(\vec{v}) = \|\vec{v}\|_1 = \sum_{i=1}^{m} |v_i| \,.$$

$$\ell_2(\vec{v}) = \|\vec{v}\| = \sqrt{\sum_{i=1}^{m} |v_i|^2} \,.$$

$$\ell_\infty(\vec{v}) = \|\vec{v}\|_\infty = \lim_{p \to \infty} \|\vec{v}\|_p = \max_{i=1}^{m} |v_i| \,.$$

Die sukzessiven Minima $\lambda_1, \lambda_2, \ldots, \lambda_n$ und die zugehörigen kürzesten linear unabhängigen Vektoren des Gitters sind abhängig von der jeweils benutzen Norm.

Beispiel: (aus [MG02, S. 8])

Gegeben seien die zweidimensionalen Basisvektoren $\vec{b}_1 = \begin{pmatrix} 2.0 \\ 0.0 \end{pmatrix}$ und $\vec{b}_2 = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}$ und das durch diese Vektoren erzeugte Gitter $\mathcal{L}(\vec{b}_1, \vec{b}_2)$.

Der Basisvektor $\vec{b}_1$ ist in der Summennorm ℓ_1 ein kürzester nichtverschwindender Vektor, da $2.0 = \|\vec{b}_1\|_1 = \|\vec{b}_2\|_1 = 2.0$ gilt. Da λ_1 nur einen Vektor enthält, ist es egal, welcher der beiden Basisvektoren für λ_1 benutzt wird. Somit ist $\lambda_1(\mathcal{L}(\vec{b}_1, \vec{b}_2)) = \|\vec{b}_1\|_1 = 2.0$.

Bei der euklidischen Norm ℓ_2 ist diese Möglichkeit nicht mehr gegeben, da $2.0 = \|\vec{b}_1\| > \|\vec{b}_2\| = \sqrt{2.0}$ ist. Somit ist $\lambda_1(\mathcal{L}(\vec{b}_1, \vec{b}_2)) = \|\vec{b}_2\| = \sqrt{2.0}$.

Bei der Maximumsnorm ℓ_∞ gilt $2.0 = \|\vec{b}_1\|_\infty > \|\vec{b}_2\|_\infty = 1.0$ und somit ist $\lambda_1(\mathcal{L}(\vec{b}_1, \vec{b}_2)) = \|\vec{b}_2\|_\infty = 1.0$.

Das Beispiel zeigt dass die Konstanten $\lambda_1, \lambda_2, \ldots, \lambda_n$ nicht eindeutig bestimmt sind. Aus diesem Grund wird im Folgenden für das sukzessive Minimum λ_1 auch von einem kürzesten Vektor und nicht von dem kürzesten Vektor gesprochen.

In den meisten Fällen wird nur die euklidische Norm ℓ_2 verwendet, die bspw. benutzt wird, um die Distanz zweier Vektoren $\vec{v}, \vec{w} \in \mathbb{R}^m$ im euklidischen Raum zu bestimmen:

$$\|\vec{v} - \vec{w}\| = \sqrt{\sum_{i=1}^{m} |v_i - w_i|^2}\,.$$

Im Folgenden bezeichnet $\|\cdot\|$ immer die euklidische Norm.

Die Abbildung 2.12 zeigt die sukzessiven Minima λ_1 und λ_2 für das zweidimensionale Beispielgitter in Abbildung 2.1. Abbildung 2.12 zeigt des Weiteren, dass die sukzessiven Minima nicht eindeutig sein müssen.

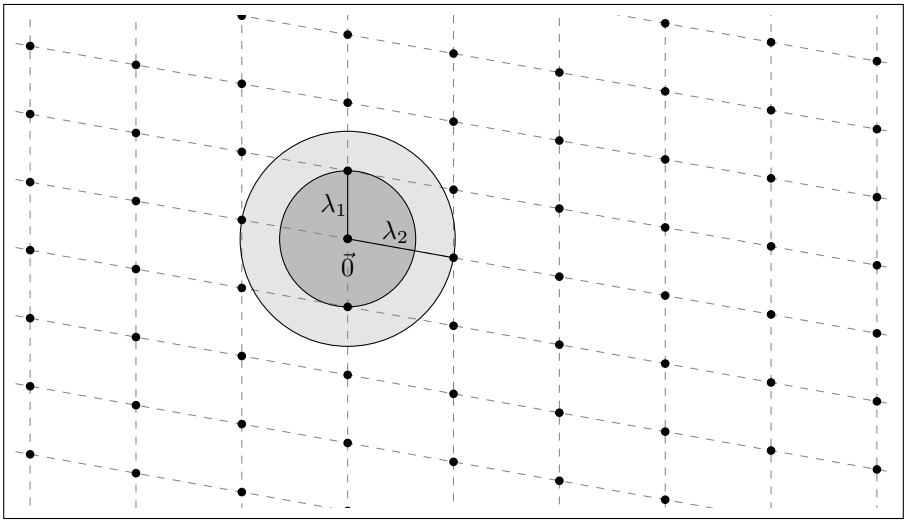


Abbildung 2.12: Die sukzessiven Minima λ_1 und λ_2 für ein zweidimensionales Beispielgitter.

Für die sukzessiven Minima $\lambda_1, \lambda_2, \ldots, \lambda_n$ gilt folgender Zusammenhang: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, da eine Kugel, die i linear unabhängige Vektoren beinhaltet, auch $i-1$ linear unab-

hängige Vektoren beinhalten muss. Aus diesem Grund muss λ_i mindestens größer gleich λ_{i-1} sein.

Satz 2.34 *Das sukzessive Minimum $\lambda_1(\mathcal{L})$ ist ein kürzester Vektor im Gitter $\mathcal{L}$ und stimmt mit der minimalen Distanz zweier beliebiger Gittervektoren überein:*

$$\lambda_1(\mathcal{L}) = \min_{\vec{v}\neq\vec{w}\in\mathcal{L}} \|\vec{v}-\vec{w}\| = \min_{\vec{v}\in(\mathcal{L}\setminus\{\vec{0}\})} \|\vec{v}\|\,.$$

Beweis: (nach [MG02, S. 9 f.])

Als Erstes muss gezeigt werden, dass λ_1 immer positiv sein muss, also $\lambda_1 > 0$ gilt:

Hilfssatz 2.35 *Sei B die Basis des Gitters $\mathcal{L}$ und O die zugehörige Orthogonalbasis zu B, die mithilfe der Gram-Schmidt Orthogonalisierung erzeugt wurde, dann gilt folgender Zusammenhang:*

$$\lambda_1 \geq \min_j \|\vec{o}_j\| > 0\,.$$

Da alle Vektoren der Orthogonalbasis, die durch die Gram-Schmidt Orthogonalisierung erzeugt werden, linear unabhängig sind, so sind diese alle verschieden vom Nullvektor $\vec{0}$. Sei $\vec{v} = x_1\vec{b}_1 + x_2\vec{b}_2 + \cdots + x_n\vec{b}_n$, wobei $x_1, x_2, \ldots, x_n \in \mathbb{Z}$ ist, ein beliebiger nichtverschwindender Vektor im Gitter und i der größte Index, sodass $x_i \neq 0$ gilt. Es wird gezeigt, dass $\|\vec{v}\| \geq \|\vec{o}_i\| \geq \min_j \|\vec{o}_j\|$ gilt. Daraus folgt, dass $\lambda_1 \geq \min_j \|\vec{o}_j\|$ gilt. Um zu zeigen, dass der Gittervektor $\vec{v}$ länger als $\vec{o}_i$ ist, muss folgender Zusammenhang gezeigt werden:

$$\sqrt{|\langle\vec{v},\vec{o}_i\rangle|} \geq \sqrt{|\langle\vec{o}_i,\vec{o}_i\rangle|}\,.$$

Durch Quadrieren entsteht:

$$|\langle\vec{v},\vec{o}_i\rangle| \geq |\langle\vec{o}_i,\vec{o}_i\rangle|\,.$$

Im Folgenden wird $\vec{v}$ alternativ geschrieben als: $\vec{v} = x_1\vec{b}_1 + x_2\vec{b}_2 + \cdots + x_i\vec{b}_i$. Die Linearkombination besteht aus nur i Vektoren, da oben i als größter Index festgelegt wurde, sodass $x_i \neq 0$ gilt. Somit ist das Skalarprodukt

$$\langle\vec{v},\vec{o}_i\rangle = \langle(x_1\vec{b}_1 + x_2\vec{b}_2 + \cdots + x_i\vec{b}_i),\vec{o}_i\rangle\,.$$

Die Summe kann aufgrund der Bilinearität zerlegt werden in:

$$= \langle x_1\vec{b}_1,\vec{o}_i\rangle + \langle x_2\vec{b}_2,\vec{o}_i\rangle + \cdots + \langle x_i\vec{b}_i,\vec{o}_i\rangle\,.$$

Aufgrund der Bilinearität können die konstanten Faktoren $x_1, x_2, \ldots, x_n$ aus dem Skalarprodukt herausgezogen werden:

$$= x_1\langle\vec{b}_1, \vec{o}_i\rangle + x_2\langle\vec{b}_2, \vec{o}_i\rangle + \cdots + x_i\langle\vec{b}_i, \vec{o}_i\rangle\,.$$

Diese Summe kann wieder mithilfe einer Summenformel vereinfacht werden:

$$= \sum_{j=1}^{i} x_j\langle\vec{b}_j, \vec{o}_i\rangle\,.$$

Die Berechnungsformel aus dem Gram-Schmidt Verfahren kann wie folgt umgeschrieben werden:

$$\vec{o}_i = \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)$$

$$\vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i) = \vec{b}_i$$

$$\vec{b}_i = \vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)\,.$$

Diese Formel kann in die Berechnung des Skalarprodukts eingesetzt werden:

$$\begin{aligned}\langle\vec{v}, \vec{o}_i\rangle &= \sum_{j=1}^{i} x_j\langle\vec{b}_j, \vec{o}_i\rangle \\ &= \sum_{j=1}^{i} x_j\langle\vec{o}_j + \sum_{k=1}^{j-1} p_{\vec{o}_k}(\vec{b_j}), \vec{o}_i\rangle\,.\end{aligned}$$

Wiederum aufgrund der Bilinearität kann dieses Skalarprodukt in mehrere Skalarprodukte zerlegt werden:

$$= \sum_{j=1}^{i} x_j \left(\langle\vec{o}_j, \vec{o}_i\rangle + \left\langle\sum_{k=1}^{j-1} p_{\vec{o}_k}(\vec{b_j}), \vec{o}_i\right\rangle\right)\,.$$

Das rechte Skalaprodukt $\langle\sum_{k=1}^{j-1} p_{\vec{o}_k}(\vec{b_j}), \vec{o}_i\rangle$ ist gleich 0, da alle Vektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{j-1}$ orthogonal zum Vektor $\vec{o}_i$ sind. Aus diesem Grund lässt sich die Formel weiter vereinfachen in:

$$= \sum_{j=1}^{i} x_j\langle\vec{o}_j, \vec{o}_i\rangle\,.$$

Aus der übrigen Summe bleibt lediglich ein Summand übrig, da wiederum die Vektoren $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{j-1}$ orthogonal zum Vektor $\vec{o}_i$ sind:

$$= x_i \langle \vec{o}_i, \vec{o}_i \rangle \,.$$

Da x_i eine von Null verschiedene Ganzzahl ist, gilt: $|\langle \vec{v}, \vec{o}_i \rangle| = |x_i \langle \vec{o}_i, \vec{o}_i \rangle| \geq |\langle \vec{o}_i, \vec{o}_i \rangle|$. Somit ist λ_1 mindestens so groß, wie der kleinste durch die Orthogonalisierung entstandene Basisvektor.

Ein Beweis, dass es mindestens einen Gittervektor mit der Länge λ_1 geben muss, befindet sich in [MG02, S. 10].

□

Folgerung 2.36 *Aus Satz 2.34 folgt, dass $\lambda_1(\mathcal{L})$ eine untere Schranke für einen kürzesten nichtverschwindenden Vektor im Gitter $\mathcal{L}$ ist. Es kann kein Vektor $\vec{v} \in \mathcal{L}$ existieren, sodass $\|\vec{v}\| < \lambda_1(\mathcal{L})$ gilt.*

Folgerung 2.29 (Satz von Minkowski) kann benutzt werden, um das sukzessive Minimum λ_1 und somit auch einen kürzesten nichtverschwindenden Vektor in einem Gitter nach oben abzuschätzen.

Folgerung 2.37 *Sei $\mathcal{L}$ ein Gitter, so gilt: Das sukzessive Minimum λ_1 bzw. ein kürzester nichtverschwindender Vektor im Gitter $\mathcal{L}$ besitzt eine Länge, die kleiner als $\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})}$ ist. Es gilt:*

$$\lambda_1 < \sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})} \,.$$

Beweis: (nach [MG02, S. 12 f.])

Sei $\mathcal{K}_0(\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L}(B))}) \cap \operatorname{span}(B)$ die n-dimensionale Kugel in der linearen Hülle des Gitters $\mathcal{L}$ mit dem Radius $\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})}$, deren Mittelpunkt der Ursprung des Gitters $\mathcal{L}$ ist. Dann besitzt diese Hyperkugel ein echt größeres Volumen als ein n-dimensionaler Würfel mit der Kantenlänge $2\sqrt[n]{\det(\mathcal{L})}$.

Dieser Hyperwürfel mit der Kantenlänge $2\sqrt[n]{\det(\mathcal{L})}$ besitzt eine maximale Diagonale von

$$\sqrt{n \cdot 2\sqrt[n]{\det(\mathcal{L})}} = \sqrt{n} \cdot \sqrt{2} \cdot \sqrt[2n]{\det(\mathcal{L})} \,.$$

Da der Mittelpunkt der Kugel $\mathcal{K}_0(\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})})$ im Ursprung liegt, muss die Hälfte dieser Diagonalen kleiner als der Radius der Kugel $\mathcal{K}_0$ sein, damit der Hyperwürfel komplett in der Kugel liegt:

$$\frac{\sqrt{n} \cdot \sqrt{2} \cdot \sqrt[2n]{\det(\mathcal{L})}}{2}$$

$$= \sqrt{n} \cdot \frac{\sqrt{2}}{2} \cdot \sqrt[2n]{\det(\mathcal{L})}$$

$$= \sqrt{n} \cdot \frac{\sqrt{2}}{\sqrt{2}\sqrt{2}} \cdot \sqrt[2n]{\det(\mathcal{L})}$$

$$= \sqrt{n} \cdot \underbrace{\frac{1}{\sqrt{2}}}_{<1} \cdot \underbrace{\sqrt[2n]{\det(\mathcal{L})}}_{<\sqrt[n]{\det(\mathcal{L})} \text{ für } n \geq 1} .$$

Daraus folgt:

$$\sqrt{n} \cdot \frac{1}{\sqrt{2}} \cdot \sqrt[2n]{\det(\mathcal{L})} < \sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})} ,$$

wobei n der Rang des Gitters $\mathcal{L}$ ist und $n \geq 1$ gilt.

Das zeigt, dass die n-dimensionale Kugel $\mathcal{K}_0(\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})})$ einen Hyperwürfel mit Kantenlänge $2\sqrt[n]{\det(\mathcal{L})}$ komplett umschließt. Der Hyperwürfel besitzt das Volumen $2^n \det(\mathcal{L})$. Daraus folgt, dass für das Volumen der Hyperkugel $\text{vol}(\mathcal{K}_0(\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})}) > 2^n \det(\mathcal{L})$ gilt.

Die Hyperkugel $\mathcal{K}_0(\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})})$ ist symmetrisch zum Ursprung des Gitters $\mathcal{L}$ und konvex. Nach Folgerung 2.29 existiert daher mindestens ein nichtverschwindender Gittervektor $\vec{v} \in (\mathcal{L} \backslash \{\vec{0}\})$, sodass $\vec{v} \in \mathcal{K}_0(\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})})$ ist und $\|v\| < \sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})}$ gilt. Dies bedeutet, dass das sukzessive Minimum λ_1 auf jeden Fall kleiner als $\sqrt{n} \cdot \sqrt[n]{\det(\mathcal{L})}$ sein muss.

□

Kapitel 3

Berechnungsprobleme

Dieses Kapitel stellt Berechnungsprobleme in Gittern vor, die u. a. zur Konstruktion von gitterbasierten Kryptografieverfahren benutzt werden können. Dabei werden einige Kenntnisse aus der Komplexitätstheorie als bekannt vorausgesetzt. Eine Einführung in die Komplexitätstheorie, sowie die Komplexitätsklassen P, NP und deren Eigenschaften gibt bspw. [Sip06].

Zu jedem Berechnungsproblem wird ein Approximationsalgorithmus angegeben, der das Problem bis auf einen nach oben beschränkten Ungenauigkeitsfaktor lösen kann. Für die Analyse der Berechnungsprobleme und eine Implementierung von darauf konstruierten gitterbasierten Kryptografieverfahren werden grundsätzlich nur rationale Gitter verwendet. Ein rationales Gitter ist ein Gitter, dessen Basis eine Matrix $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ ist. Da jedes rationale Gitter in ein ganzzahliges Gitter (mit Basis $B \in \mathbb{Z}^{m \times n}$) transformiert werden kann [MG02, S. 5], könnten theoretisch nur ganzzahlige Gitter betrachtet werden. Dies vereinfacht bestimmte Eigenschaften so stark, dass in diesem Kapitel generell rationale Gitter bzw. entsprechende Gitterprobleme in solchen Gittern untersucht werden.

Eine Analyse der „Schwierigkeit“ dieser Berechnungsprobleme folgt im anschließenden Kapitel.

3.1 Effiziente Gitterprobleme

Als Erstes werden effiziente Gitterprobleme vorgestellt. Für die folgenden Berechnungsprobleme existiert somit jeweils ein Algorithmus in P, der dieses Berechnungsproblem exakt lösen kann.

Definition 3.1 *Das Problem der Gitterzugehörigkeit (MEMBERSHIP) eines Vektors ist wie folgt definiert:*

PROBLEM: *MEMBERSHIP*
EINGABE: *Eine Gitterbasis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$ und ein Vektor $\vec{q} \in \text{span}(B)$.*
FRAGE: *Ist $\vec{q} \in \text{span}(B)$ ein Gittervektor, also gilt $\vec{q} \in \mathcal{L}$?*

Satz 3.2 *MEMBERSHIP ist in P.*

Beweis:

Als Beweis reicht die Angabe eines Polynomialzeitalgorithmus für MEMBERSHIP.

Um entscheiden zu können, ob ein Vektor $\vec{q} \in \text{span}(B)$ ein Gittervektor ist, muss folgendes lineares Gleichungssystem gelöst werden:

$$B \cdot \vec{x} = \vec{q},$$

wobei $B \in \mathbb{Q}^{m \times n}$ eine Gitterbasis und $\vec{x} \in \mathbb{Q}^n$ ein n-dimensionaler Vektor ist. Dieser n-dimensionale Vektor $\vec{x}$ beschreibt die Linearkombination, die benötigt wird, um den Vektor $\vec{q}$ aus den Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$, die in der Matrix B als Spaltenvektoren enthalten sind, zu beschreiben. Der Vektor $\vec{q} \in \text{span}(B)$ ist genau dann ein Gittervektor, wenn $\vec{x} \in \mathbb{Z}^n$ gilt, also alle n Komponenten des Vektors $\vec{x}$ ganzzahlig sind. Genau in diesem Fall kann der Vektor $\vec{q}$ als ganzzahlige Linearkombination der Basisvektoren der Gitterbasis beschrieben werden, was der Definition eines Gittervektors entspricht.

Dieses lineare Gleichungssystem kann mithilfe des Gaußschen Eliminationsverfahren in eine Zeilenstufenform transformiert und gelöst werden. Der Rechenaufwand des Gaußschen Eliminationsverfahren ist in $\mathcal{O}(m^3)$ enthalten, wobei m die Dimension des Gitters angibt.

□

Definition 3.3 *Das Problem der Längenbestimmung (LENGTH) eines Gittervektors ist wie folgt definiert:*

PROBLEM: *LENGTH*
EINGABE: *Ein Gittervektor $\vec{v} \in \mathcal{L}$.*
AUSGABE: *Die euklidische Länge $\|\vec{v}\|$ des Gittervektors $\vec{v} \in \mathcal{L}$.*

Satz 3.4 *LENGTH ist in P.*

Beweis:

Als Beweis reicht die Angabe eines Polynomialzeitalgorithmus für LENGTH.

Die Länge eines Vektors $\vec{v}$ kann wie folgt berechnet werden:

$$\|\vec{v}\| = \sqrt{\sum_{i=1}^{m} |v_i|^2}\,.$$

Dies zeigt, dass der Rechenaufwand zur Bestimmung von $\|\vec{v}\|$ in $\mathcal{O}(m)$ enthalten ist, wobei m die Dimension des Gitters angibt.

□

Definition 3.5 *Das Problem der Distanzbestimmung (DISTANCE) eines Gittervektors und eines Vektors in der linearen Hülle des Gitters ist wie folgt definiert:*

PROBLEM: *DISTANCE*
EINGABE: *Ein Gittervektor $\vec{v} \in \mathcal{L}$ und ein Vektor $\vec{q} \in \mathrm{span}(B)$.*
AUSGABE: *Die euklidische Länge $\|\vec{v} - \vec{q}\|$ zwischen dem Gittervektor $\vec{v} \in \mathcal{L}$ und dem Vektor $\vec{q} \in \mathrm{span}(B)$.*

Satz 3.6 *DISTANCE ist in P.*

Beweis:

Als Beweis reicht die Angabe eines Polynomialzeitalgorithmus für DISTANCE.

Die Distanz zwischen zwei Vektoren kann wie folgt berechnet werden:

$$\|\vec{v} - \vec{q}\| = \sqrt{\sum_{i=1}^{m} |v_i - q_i|^2}\,.$$

Dies zeigt, dass der Rechenaufwand zur Bestimmung von $\|\vec{v} - \vec{q}\|$ in $\mathcal{O}(m)$ enthalten ist, wobei m die Dimension des Gitters angibt.

□

Definition 3.7 *Das Problem der Gitteräquivalenz (EQUIVALENCE) zweier Basen $B, B' \in \mathbb{Q}^{m \times n}$ ist wie folgt definiert:*

PROBLEM: *EQUIVALENCE*
EINGABE: *Zwei Basen $B, B' \in \mathbb{Q}^{m \times n}$ mit $m \geq n$.*
FRAGE: *Erzeugen $B, B' \in \mathbb{Q}^{m \times n}$ ein äquivalentes Gitter $\mathcal{L}$, also gilt $\mathcal{L}(B) = \mathcal{L}(B')$?*

Satz 3.8 *EQUIVALENCE ist in P.*

Beweis:

Als Beweis reicht die Angabe eines Polynomialzeitalgorithmus für EQUIVALENCE.

Um entscheiden zu können, ob die beiden Basen $B, B' \in \mathbb{Q}^{m \times n}$ ein äquivalentes Gitter erzeugen, muss folgendes lineares Gleichungssystem gelöst werden:

$$B' = B \cdot M \,,$$

wobei $M \in \mathbb{Q}^{n \times n}$ eine quadratische Matrix ist. Gilt $M \in \mathbb{Z}^{n \times n}$ und $|\det(M)| = 1$, so ist die Matrix M unimodular und die beiden Basen $B, B' \in \mathbb{Q}^{m \times n}$ erzeugen ein äquivalentes Gitter.

Auch dieses lineare Gleichungssystem kann mithilfe des Gaußschen Eliminationsverfahren gelöst werden, dessen Rechenaufwand in $\mathcal{O}(m^3)$ enthalten ist, wobei m die Dimension des Gitters angibt. Der Rechenaufwand für die Bestimmung der Matrixdeterminanten einer quadratischen $n \times n$-Matrix ist in $\mathcal{O}(n^3)$ enthalten, sodass der Rechenaufwand für EQUIVALENCE in $\mathcal{O}(m^3) + \mathcal{O}(n^3) \underbrace{=}_{\text{da } m \geq n} \mathcal{O}(2m^3) = \mathcal{O}(m^3)$ enthalten ist.

□

Definition 3.9 *Das Problem der Bestimmung eines dualen Gitters $\mathcal{L}^*$ zu einem Gitter $\mathcal{L}$ (DUAL) ist wie folgt definiert:*

PROBLEM: *DUAL*
EINGABE: *Ein Gitter $\mathcal{L}(B)$ mit $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$.*
AUSGABE: *Ein Gitter $\mathcal{L}^*$, sodass $\langle \vec{v}, \vec{w} \rangle \in \mathbb{Z}$ für alle Vektoren Gittervektoren $\vec{v} \in \mathcal{L}$ und $\vec{w} \in \mathcal{L}^*$.*

Satz 3.10 *DUAL ist in P.*

Beweis:

Als Beweis reicht die Angabe eines Polynomialzeitalgorithmus für DUAL. Aus Satz 2.17 folgt, dass $\mathcal{L}^*(B) = \mathcal{L}(D)$ mit der Basis $D = B(B^T B)^{-1} \in \mathbb{Q}^{m \times n}$ das duale Gitter zu $\mathcal{L}(B)$ erzeugt. Für die Bestimmung eines dualen Gitters muss demnach eine $m \times n$-Matrix transponiert ($\mathcal{O}(m^2)$), mit einer anderen Matrix multipliziert ($\mathcal{O}(m^3)$), diese invertiert ($\mathcal{O}(m^3)$) und abschließend mit einer anderen Matrix wiederum multipliziert ($\mathcal{O}(m^3)$) werden. Für den Rechenaufwand die Bestimmung einer zur Gitterbasis B duale Gitterbasis ergibt sich:

$$\mathcal{O}(m^2) + \mathcal{O}(m^3) + \mathcal{O}(m^3) + \mathcal{O}(m^3) = \mathcal{O}(4m^3) = \mathcal{O}(m^3) \,.$$

Damit ist DUAL in $\mathcal{O}(m^3)$ und somit auch in der Komplexitätsklasse P enthalten.

□

Da die bisher vorgestellten Gitterprobleme in P liegen, können diese nicht zum sicheren Verschlüsseln von Informationen benutzt werden. Die vorgestellten Probleme zeigen, welche Gitterprobleme bspw. zur Überprüfung einer Signatur, die mithilfe von gitterbasierten Kryptografieverfahren erstellt wurde, benutzt werden können.

Im Folgenden werden „schwierige“ Berechnungsprobleme in Gittern vorgestellt. Der Begriff „schwierig“ bedeutet in diesem Kontext, dass keine Polynomialzeitalgorithmen bekannt sind, die diese Probleme exakt lösen können.

3.2 „Closest Vector Problem“

Definition 3.11 *Das „Closest Vector Problem (CVP)“ ist wie folgt definiert:*

PROBLEM: *CVP*

EINGABE: *Eine Basis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$ und ein Vektor $\vec{q} \in \text{span}(B)$.*

AUSGABE: *Ein Gittervektor $\vec{v} \in \mathcal{L}$, sodass gilt:*

$$\|\vec{q} - \vec{v}\| \leq \|\vec{q} - \vec{w}\|$$

für alle Gittervektoren $\vec{w} \in (\mathcal{L} \backslash \{\vec{v}\})$.

Die Lösung des „Closest Vector Problem“ ist demnach der nächste Gittervektor zu einem vorgegebenen Vektor, der in der linearen Hülle des Gitters liegt.

Anmerkung: Es existieren Fälle, in dem der nächste Gittervektor $\vec{v} \in \mathcal{L}$ zu einem Vektor $\vec{q} \in \text{span}(B)$ nicht eindeutig bestimmt ist. Es können mehrere Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_i \in \mathcal{L}$ existieren, sodass $\|\vec{q} - \vec{v}_1\| = \|\vec{q} - \vec{v}_2\| = \cdots = \|\vec{q} - \vec{v}_i\|$ gilt.

Die Abbildung 3.1 verdeutlicht das „Closest Vector Problem“ anhand eines zweidimensionalen Beispiels. In diesem Beispiel wird der nächste Gittervektor zum Vektor $\vec{q} \in \mathbb{R}^2$ gesucht.

Da für das „Closest Vector Problem“ derzeit kein Polynomialzeitalgorithmus bekannt ist, der dieses Problem exakt lösen kann, wird noch die Approximationsvariante dieses Problems untersucht. Das zugehörige Approximationsproblem ist wie folgt definiert:

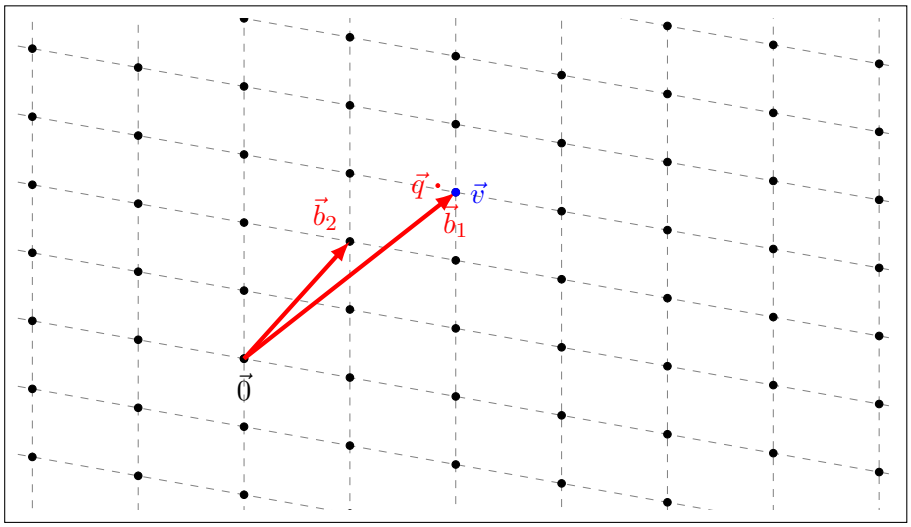


Abbildung 3.1: Ein Beispiel für das „Closest Vector Problem“ in $\mathbb{R}^2$.

Definition 3.12 *Das „Approximate Closest Vector Problem ($\mathsf{CVP}_{\gamma(n)}$)“ für $\gamma(n) \geq 1$ ist wie folgt definiert:*

PROBLEM: $\mathsf{CVP}_{\gamma(n)}$
EINGABE: *Eine Basis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$, ein Vektor $\vec{q} \in \text{span}(B)$.*
AUSGABE: *Ein Gittervektor $\vec{v} \in \mathcal{L}$, sodass gilt:*

$$\|\vec{q} - \vec{w}\| \leq \|\vec{q} - \vec{v}\| \leq \gamma(n) \cdot \|\vec{q} - \vec{w}\|,$$

wobei $\vec{w} \in \mathcal{L}$ ein nächster Gittervektor zu $\vec{q}$ ist.

Bei diesem Approximationsproblem $\mathsf{CVP}_{\gamma(n)}$ ist $\gamma(n) \geq 1$ eine Funktion für die Güte der berechneten Lösung im Vergleich zur optimalen Lösung in Abhängigkeit von n, wobei n in vielen Fällen dem Rang des Gitters entspricht. Die Abhängigkeit der Approximationsfunktion $\gamma(n)$ versucht die Güte der approximierten Lösung in Gittern mit unterschiedlichen Rängen vergleichbar zu machen.

Für das „Approximate Closest Vector Problem“ hat László Babai 1986 mehrere Heuristiken angegeben [Bab86]. Eine einfache Heuristik ist Babais ROUNDING OFF-PROCEDURE.

Hinweis: Der Begriff „rounding off“ bezeichnet in der englischen Sprache eine Rundung und nicht, wie angenommen werden könnte, das Abrunden einer Zahl.

Anmerkung: Im Folgenden ist mit $\lfloor r \rceil$ für $r \in \mathbb{R}$ die ganze Zahl $x \in \mathbb{Z}$ gemeint, sodass $|r - x|$ minimal ist. Liegt die Zahl r genau zwischen zwei ganzen Zahlen x und $x+1$, sodass $|r - x| = |r - (x+1)|$ gilt, dann wird r zur nächsten geraden ganzen Zahl gerundet. Analog

kann $\lfloor r \rceil$ für $r \in \mathbb{R}$ auch wie folgt definiert werden:

$$\lfloor r \rceil =_{\text{def}} \begin{cases} \lfloor r + 0.5 \rfloor \text{, falls } r - \lfloor r \rfloor \neq \frac{1}{2} \\ \lfloor r \rfloor \text{, falls } r - \lfloor r \rfloor = \frac{1}{2} \text{ und } \lfloor r \rfloor = 2k \text{ mit } k \in \mathbb{Z} \\ \lceil r \rceil \text{, falls } r - \lfloor r \rfloor = \frac{1}{2} \text{ und } \lceil r \rceil = 2k \text{ mit } k \in \mathbb{Z}\,. \end{cases}$$

Es gilt $|r - \lfloor r \rceil| \leq \frac{1}{2}$ für alle $r \in \mathbb{R}$. $\lfloor r \rceil$ entspricht somit dem mathematischen Runden.

Beispiel: Es werden kurz mehrere Beispiele für diese Rundungsprozedur angegeben, damit die Bedeutung dieser Schreibweise deutlich wird.

$$\begin{aligned} \lfloor 3.14 \rceil &= \lfloor 3.14 + 0.5 \rfloor = \lfloor 3.64 \rfloor = 3 \\ \lfloor -2.71 \rceil &= \lfloor -2.71 + 0.5 \rfloor = \lfloor -2.21 \rfloor = -3 \\ \lfloor 0.5 \rceil &= \lfloor 0.5 \rfloor = 0 \\ \lfloor 1.5 \rceil &= \lceil 1.5 \rceil = 2\,. \end{aligned}$$

Der Algorithmus 3.1 approximiert das „Closest Vector Problem“. Babais ROUNDING

Algorithmus 3.1 : Babais ROUNDING OFF-PROCEDURE

Eingabe : Eine Basis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$ und ein Vektor $\vec{q} \in \text{span}(B)$.
Ausgabe : Ein Gittervektor $\vec{v} \in \mathcal{L}$.

```
1 Berechne r_1, r_2, ..., r_n ∈ ℝ, sodass q = r_1 b_1 + r_2 b_2 + ... + r_n b_n gilt.
2 v ← 0
3 for i = 1, 2, ..., n do
4 |   v ← v + ⌊r_i⌉ · b_i
5 end for
```

OFF-PROCEDURE ist eine mathematische Rundung von n reellen Zahlen. Zuerst wird der Vektor $\vec{q} \in \text{span}(B)$ als reelle Linearkombination der Basisvektoren des Gitters beschrieben. Die reellen Koeffizienten dieser Linearkombination werden dann einfach zur nächsten Ganzzahl auf- bzw. abgerundet. Die Güte des vom Approximationsalgorithmus 3.1 berechneten Gittervektors $\vec{v} \in \mathcal{L}$ mit $\vec{v} = \lfloor r_1 \rceil \vec{b}_1 + \lfloor r_2 \rceil \vec{b}_2 + \ldots + \lfloor r_n \rceil \vec{b}_n$ und $\lfloor r_1 \rceil, \lfloor r_2 \rceil, \ldots, \lfloor r_n \rceil \in \mathbb{Z}$ hängt im Wesentlichen von der eingegebenen Basis $B \in \mathbb{Q}^{m \times n}$ des Gitters ab.

Anmerkung: Algorithmus 3.1 findet einen Gittervektor, der maximal um einen exponentiellen Faktor von der exakten Lösung entfernt ist, sofern die eingegebene Gitterbasis $B \in \mathbb{Q}^{m \times n}$ LLL-reduziert ist. Dieser Begriff wird später in diesem Kapitel erläutert.

Für die Güte des gefundenen Gittervektors, gilt allgemein folgender Zusammenhang: Sind $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ Basisvektoren einer „schlechten“ (langen und nichtorthogonalen) Basis, so ist auch der ausgebenene Gittervektor $\vec{v} \in \mathcal{L}$ eine schlechte Annäherung an die exakte Lösung des „Closest Vector Problems“, da der berechnete Gittervektor $\vec{v}$ eine ganzzahlige Linearkombination dieser schlechten Basisvektoren ist. Der ausgegebene Vektor kann daher weit von der exakten Lösung entfernt sein.

Je höher die Dimension des Gitters ist, desto ungenauer ist die Abschätzung durch Babais Heuristik bei einer solchen schlechten Basis [HPS08, S. 380].

Für Babais ROUNDING OFF-PROCEDURE existieren Spezialfälle, in denen diese Heuristik den nächsten Gittervektor zu einem gegebenen Vektor exakt berechnen kann.

Definition 3.13 *Sei $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ die Basis eines Gitters $\mathcal{L}$, dann ist das* Hadamard-Verhältnis *wie folgt definiert:*

$$\mathcal{H}(B) = \frac{\det(\mathcal{L})}{\prod_{i=1}^{n} \|\vec{b}_i\|}$$

Da $\det(\mathcal{L}) \leq \prod_{i=1}^{n} \|\vec{b}_i\|$ (Hadamard-Ungleichung) gilt und die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ linear unabhängig sind, gilt $0 < \mathcal{H}(B) \leq 1$. Je dichter $\mathcal{H}(B)$ am Wert 1 ist, desto orthogonaler sind die Basisvektoren der Basis B. Es gilt genau dann $\mathcal{H}(B) = 1$, wenn B eine Orthogonalbasis ist. Nicht für jedes Gitter existiert eine Orthogonalbasis, deren Basisvektoren alle im Gitter enthalten sind.

Satz 3.14 *Sei $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eine Orthogonalbasis eines Gitters $\mathcal{L}$ und $\vec{q} \in \operatorname{span}(B)$ ein beliebiger Vektor mit*

$$\vec{q} = r_1\vec{b}_1 + r_2\vec{b}_2 + \cdots + r_n\vec{b}_n \text{ und } r_1, r_2, \ldots, r_n \in \mathbb{R},$$

dann ist der Gittervektor

$$\vec{v} = \lfloor r_1 \rceil \vec{b}_1 + \lfloor r_2 \rceil \vec{b}_2 + \cdots + \lfloor r_n \rceil \vec{b}_n \text{ und } \lfloor r_1 \rceil, \lfloor r_2 \rceil, \ldots, \lfloor r_n \rceil \in \mathbb{Z}$$

der nächste Gittervektor zu $\vec{q}$.

Beweis:

Damit der Gittervektor $\vec{v} \in \mathcal{L}(B)$ der nächste Gittervektor zum Vektor $\vec{q} \in \operatorname{span}(B)$ ist, muss $\|\vec{q} - \vec{v}\|$ minimal sein.

Für die euklidische Norm des Differenzvektors $\vec{q} - \vec{v}$ gilt:

$$\begin{aligned}\|\vec{q} - \vec{v}\| &= \sqrt{\langle(\vec{q} - \vec{v}), (\vec{q} - \vec{v})\rangle} = \sqrt{\langle\vec{q}, (\vec{q} - \vec{v})\rangle - \langle\vec{v}, (\vec{q} - \vec{v})\rangle} \\ &= \sqrt{\langle\vec{q}, \vec{q}\rangle - \langle\vec{q}, \vec{v}\rangle - \langle\vec{v}, \vec{q}\rangle + \langle\vec{v}, \vec{v}\rangle}\,.\end{aligned}$$

Da aufgrund der Bilinearität des Skalarprodukts in $\mathbb{Q}^m$ für $\langle\vec{q}, \vec{v}\rangle = \langle\vec{v}, \vec{q}\rangle$ gilt, folgt weiter:

$$= \sqrt{\langle\vec{q}, \vec{q}\rangle - 2\langle\vec{q}, \vec{v}\rangle + \langle\vec{v}, \vec{v}\rangle} = \sqrt{\langle\vec{v}, \vec{v}\rangle - 2\langle\vec{q}, \vec{v}\rangle + \langle\vec{q}, \vec{q}\rangle}\,.$$

Da der Gittervektor $\vec{v}$ und der Vektor $\vec{q}$ Linearkombinationen der Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ sind, gilt:

$$= \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \langle\vec{b_i}, \vec{b_j}\rangle - 2\sum_{i=1}^{n}\sum_{j=1}^{n} r_i x_j \langle\vec{b_i}, \vec{b_j}\rangle + \sum_{i=1}^{n}\sum_{j=1}^{n} r_i r_j \langle\vec{b_i}, \vec{b_j}\rangle}$$

mit $x_k \in \mathbb{Z}$ und $r_k \in \mathbb{R}$ für $k \in \{1, 2, \ldots, n\}$. Da die Basis nach dem zu beweisenden Satz eine Orthogonalbasis ist, sind alle Basisvektoren $\vec{b_i}$ und $\vec{b_j}$ für $i, j \in \{1, 2, \ldots, n\}$ und $i \neq j$ paarweise orthogonal und daraus folgt:

$$\begin{aligned}&= \sqrt{\sum_{i=1}^{n} x_i x_i \langle\vec{b_i}, \vec{b_i}\rangle - \sum_{i=1}^{n} 2 r_i x_i \langle\vec{b_i}, \vec{b_i}\rangle + \sum_{i=1}^{n} r_i r_i \langle\vec{b_i}, \vec{b_i}\rangle} \\ &= \sqrt{\sum_{i=1}^{n} x_i^2 \|\vec{b_i}\|^2 - \sum_{i=1}^{n} 2 r_i x_i \|\vec{b_i}\|^2 + \sum_{i=1}^{n} r_i^2 \|\vec{b_i}\|^2} = \sum_{i=1}^{n} (x_i - r_i)^2 \|\vec{b_i}\|^2\,.\end{aligned}$$

Da alle Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ linear unabhängig sind, folgt daraus, dass die Differenzen $(x_i - r_i)$ für $i \in \{1, 2, \ldots, n\}$ minimal sein müssen, damit der gesamte Term minimal ist. Dies ist genau dann der Fall, wenn x_i die nächste Ganzzahl zu r_i ist, wenn $x_i = \lfloor r_i \rceil$ gilt.

□

Babais Heuristik bedeutet geometrisch Folgendes: Durch das Runden der reellen Koeffizienten der Linearkombination wird die jeweilige Grundmasche des Gitter bzw. der entsprechenden Gitterbasis um den Vektor $\vec{q}$ aufgespannt, sodass der Vektor $\vec{q}$ der Mittelpunkt dieser verschobenen Grundmasche ist. Die Lösung von Babais Algorithmus ist der Gitterpunkt, der in dieser verschobenen Grundmasche enthalten ist. Es ist stets garantiert, dass eine solche verschobene Grundmasche genau einen Gittervektor enthält [Gal12, S. 372].

Die Abbildung 3.2 verdeutlicht anhand von zweidimensionalen Beispielen ($m = n = 2$), was Algorithmus 3.1 geometrisch bedeutet und warum der Fehler dieser Heuristik von der gewählten Basis bzw. der zugehörigen Grundmasche abhängig ist.

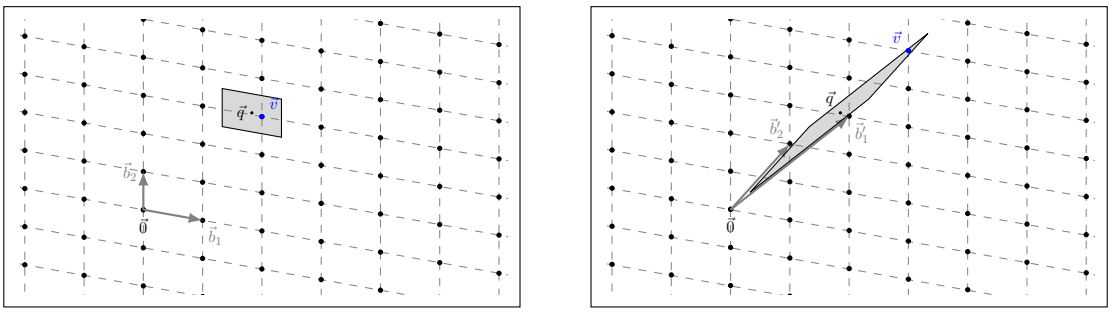


Abbildung 3.2: Babais Algorithmus für eine „nahezu orthogonale“ und eine nichtorthogonale Basis.

Die zweite Heuristik NEAREST PLANE-PROCEDURE, die László Babai in [Bab86] angegeben hat, ist komplizierter als die ROUNDING OFF-PROCEDURE. Diese Heuristik reduziert das „Closest Vector Problem“ in einem Gitter mit Rang n auf ein „Closest Vector Problem“ in einem Teilgitter mit Rang $n-1$. Diese Reduktion wird rekursiv solange druchgeführt, bis das reduzierte Gitter den Rang $n = 1$ besitzt. In diesem Gitter wird approximativ ein nächster Gittervektor gefunden und auf dieser Lösung werden aufbauend die nächsten Gittervektoren in den höheren Gittern approximiert. Der rekursive Algorithmus 3.2 zeigt, wie die NEAREST PLANE-PROCEDURE das „Closest Vector Problem“ approximiert. Zur Anwendung dieses Algorithmus muss zuerst zu der eingegebenen Basis des Gitters die zugehörige Gram-Schmidt Orthogonalbasis berechnet werden.

Algorithmus 3.2 : Babais NEAREST PLANE-PROCEDURE (nach [Gal12, S. 370])

Eingabe : Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ eines Gitters $\mathcal{L}$, die zugehörige Gram-Schmidt Orthogonalbasis $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{Q}^m$ und ein Vektor $\vec{q} \in \text{span}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$.
Ausgabe : Ein Gittervektor $\vec{v} \in \mathcal{L}$.

function nearestPlane($\{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\}, \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n\}, \vec{q}$)
begin
  $r \leftarrow \frac{\langle \vec{q}, \vec{o}_i \rangle}{\langle \vec{o}_i, \vec{o}_i \rangle}$
  $\vec{w} \leftarrow \lfloor r \rceil \cdot \vec{b}_i$
  if $n > 1$ **then**
    $\vec{q}' \leftarrow \vec{q} - (r - \lfloor r \rceil) \cdot \vec{o}_i$
    $\vec{q}'' \leftarrow \vec{q}' - \vec{w}$
    return nearestPlane($\{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{n-1}\}, \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{n-1}\}, \vec{q}''$)$+\vec{w}$
  else
    return $\vec{w}$
  end if
end

Im Folgenden wird die Reduktion des „Closest Vector Problems“ in einem Gitter mit Rang n auf ein „Closest Vector Problem“ in einem Teilgitter mit Rang $n-1$ erläutert. Die einzelnen Schritte dieser Reduktion werden wiederum anhand eines zweidimensionalen Beispiels (siehe Abbildung 3.3) veranschaulicht.

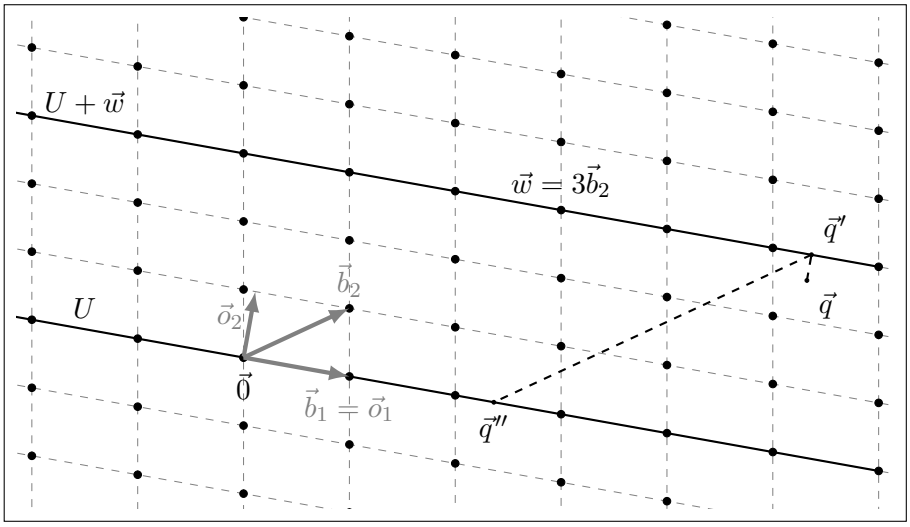


Abbildung 3.3: Babais NEAREST PLANE-Algorithmus (nach [Gal12, S. 370]).

Sei $\vec{q} \in \text{span}(B)$ der Vektor, zu dem ein nahegelgener Gittervektor im Gitter $\mathcal{L}(B)$ gefunden werden soll, $U = \text{span}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{n-1})$ der Untervektorraum, der durch die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{n-1}$ aufgespannt wird und $\mathcal{L}_U(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{n-1}) = \mathcal{L} \cap U$ das zugehörige Teilgitter mit Rang $n-1$, das im Untervektorraum U eingebettet ist. Die NEAREST PLANE-PROCEDURE approximiert einen um einen Gittervektor $\vec{w} \in \mathcal{L}$ verschobenen Untervektorraum $U + \vec{w}$, sodass die Distanz zwischen $\vec{q}$ und dem verschobenen Untervektorraum $U + \vec{w}$ minimal ist. Dieser Gittervektor $\vec{w}$ wird wie folgt approximiert. Der Vektor $\vec{q} \in \text{span}(B)$ wird als reelle Linearkombination der Gram-Schmidt orthogonalisierten Basisvektoren dargestellt, also gilt:

$$\vec{q} = r_1\vec{o}_1 + r_2\vec{o}_2 + \cdots + r_n\vec{o}_n, \text{ mit } r_1, r_2, \ldots, r_n \in \mathbb{R}\,.$$

Der Gittervektor $\vec{w} \in \mathcal{L}$ wird dann als $\vec{w} = \lfloor r_n \rceil \cdot \vec{b}_n$ approximiert. Danach wird der Vektor $\vec{q}$ auf einen Vektor $\vec{q}'$ im Untervektorraum $U + \vec{w}$ projiziert. Der Vektor $\vec{q}'$ entspricht nach dieser orthogonalen Projektion auf den Untervektorraum $U + \vec{w}$ der Form $\vec{q}' = \sum_{i=1}^{n-1} r_i\vec{o}_i + \lfloor r_n \rceil \vec{o}_n$.
Durch die Subtraktion des Gittervektors $\vec{w} \in \mathcal{L}$ vom Vektor $\vec{q}' \in U + \vec{w}$ wird ein dritter Vektor $\vec{q}''$ gebildet, der im Untervektorraum U liegt. Wenn $\vec{q}$ nicht im Gitter enthalten ist, dann ist auch auf den Untervektorraum U abgebildete Vektor $\vec{q}''$ nicht im Gitter

enthalten [Gal12, S. 366]. Der Vektor $\vec{q}''$ ist eine Instanz für das „Closest Vector Problem“ im Teilgitter $\mathcal{L}_U$ mit Rang $n-1$. Dieses Vorgehen kann rekursiv für diese Probleminstanz angewendet werden, was eine Lösung $\vec{w}' \in \mathcal{L}_U$ ergibt. Die Lösung für das Gitter $\mathcal{L}$ mit Rang n ist entsprechend $\vec{v} = \vec{w}' + \vec{w}$.

Die Güte des vom Algorithmus 3.2 berechneten Gittervektors ist wiederum maximal um einen exponentiellen Faktor schlechter, als die exakte Lösung der entsprechenden Instanz des „Closest Vector Problems“, sofern die eingegebene Gitterbasis $B \in \mathbb{Q}^{m \times n}$ LLL-reduziert ist.

Bei der ROUNDING OFF-PROCEDURE muss nicht die Gram-Schmidt Orthogonalbasis berechnet werden, dafür ist die Güte des gefundenen Gittervektora wahrscheinlich schlechter als ein gefundener Gittervektor der NEAREST PLANE-PROOCEDURE. Eine detaillierte Analyse dieser beiden vorgestellten Heuristiken für das „Closest Vector Problem“ findet sich in [Bab86].

3.3 „Shortest Vector Problem“

Neben dem „Closest Vector Problem“ werden noch weitere Probleme untersucht, auf denen asymmetrische Kryptografieverfahren basieren können. Eines dieser Berechnungsprobleme ist das sogenannte „Shortest Vector Problem“:

Definition 3.15 *Das „Shortest Vector Problem (SVP)“ ist wie folgt definiert:*

PROBLEM: *SVP*
EINGABE: *Eine Basis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$.*
AUSGABE: *Ein kürzester nichtverschwindender Gittervektor $\vec{v} \in \mathcal{L}$, sodass gilt:*

$$\|\vec{v}\| = \lambda_1 .$$

Die Abbildung 3.4 zeigt ein Beispiel für das „Shortest Vector Problem“ in einem zweidimensionalen Gitter.

Da für das „Shortest Vector Problem“ derzeit auch kein Polynomialzeitalgorithmus bekannt ist, der dieses Problem exakt lösen kann, wird auch hier zusätzlich die Approximationsvariante dieses Problems untersucht. Das zugehörige Approximationsproblem ist wie folgt definiert:

Definition 3.16 *Das „Approximate Shortest Vector Problem ($SVP_{\gamma(n)}$)“ ist wie folgt definiert:*

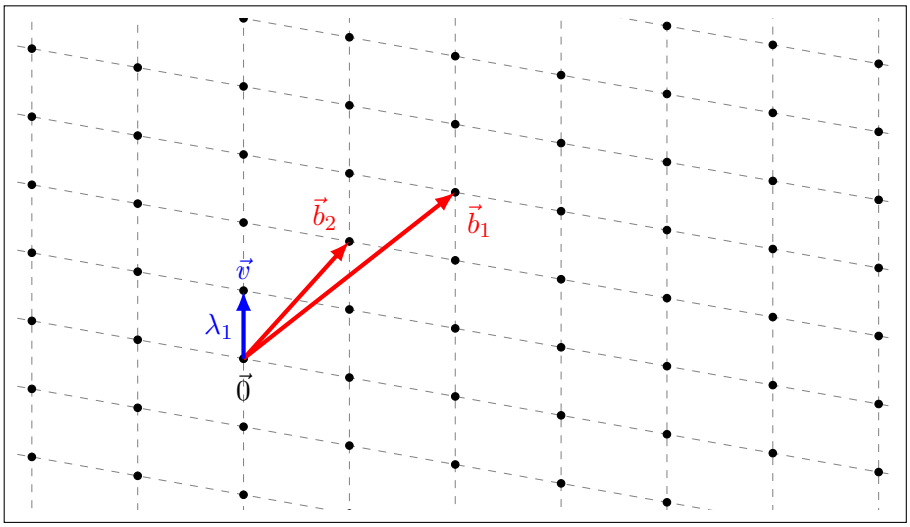


Abbildung 3.4: Das „Shortest Vector Problem“ in $\mathbb{R}^2$.

PROBLEM: $SVP_{\gamma(n)}$
EINGABE: *Eine Basis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$.*
AUSGABE: *Ein kürzester nichtverschwindender Gittervektor $\vec{v} \in \mathcal{L}$ sodass gilt:*

$$\lambda_1 \leq \|\vec{v}\| \leq \gamma(n) \cdot \lambda_1 \,.$$

Bei diesem Approximationsproblem $\mathsf{SVP}_{\gamma(n)}$ ist $\gamma(n) \geq 1$ ebenfalls eine Funktion für die Güte der berechneten Lösung im Vergleich zur optimalen Lösung in Abhängigkeit von n, wobei n in vielen Fällen dem Rang des Gitters entspricht.

Ähnlich wie beim „Closest Vector Problem“ existieren Spezialfälle, in dem das „Shortest Vector Problem“ leicht gelöst werden kann.

Satz 3.17 *Sei $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ die Orthogonalbasis eines Gitters $\mathcal{L}$, dann ist $\min_{i=1}^{n} \|\vec{b}_i\|$ eine Lösung für das „Shortest Vector Problem (SVP)“ im Gitter $\mathcal{L}$.*

Beweis:

Als Erstes wird gezeigt, dass es keine Linearkombination der Basisvektoren geben kann, die kürzer ist, als eine der paarweise orthogonalen Basisvektoren. Diese Aussage wird nur für zwei beliebige Basisvektoren $\vec{b}_i, \vec{b}_j \in \mathbb{Q}^m$ und $\langle \vec{b}_i, \vec{b}_j \rangle = 0$ gezeigt.

Da die Vektoren $\vec{b}_i$ und $\vec{b}_j$ orthogonal sind, erzeugen $\vec{b}_i, \vec{b}_j$ und $\vec{b}_i + \vec{b}_j$ ein rechtwinkliges Dreieck. Die Abbildung 3.5 veranschaulicht dies an einem zweidimensionalen Beispiel.

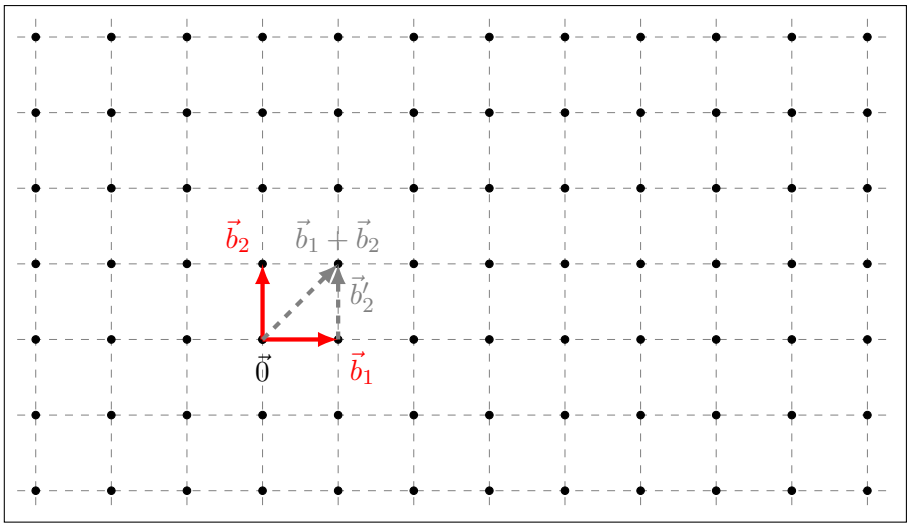


Abbildung 3.5: Eine beliebige Linearkombination von Basisvektoren einer Orthogonalbasis in $\mathbb{R}^2$.

Für die Seitenlängen dieses rechtwinkligen Dreiecks gilt nach dem Satz des Pythagoras:

$$\|\vec{b}_i\|^2 + \|\vec{b}_j\|^2 = \|\vec{b}_i + \vec{b}_j\|^2$$

Da die Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ nichtverschwindend und linear unabhängig sind, muss $\|\vec{b}_i\| < \|\vec{b}_i + \vec{b}_j\|$ und $\|\vec{b}_j\| < \|\vec{b}_i + \vec{b}_j\|$ gelten. Somit kann es keine Linearkombination $x\vec{b}_i + y\vec{b}_j$ mit $x, y \in (\mathbb{Z}\backslash\{0\})$ geben, dessen Länge kleiner als die der Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ sind. Diese Argumentation funktioniert nur bei Orthogonalbasen, da die Basisvektoren und die zugehörigen Linearkombinationen der Basisvektoren ein rechtwinkliges Dreieck aufspannen.

Als Letztes wird gezeigt, dass es bei einer Orthogonalbasis keine kürzeren Gittervektoren als die Basisvektoren geben kann:

Es soll angenommen werden, dass ein kürzerer Gittervektor $\vec{v} \in \mathcal{L}(B)$ existiert, sodass $\|\vec{v}\| < \|\vec{b}_i\|$ gilt, wobei $\vec{v}$ parallel zu $\vec{b}_i$ ist. Die Abbildung 3.6 veranschaulicht einen solchen Vektor in einem zweidimensionalen Beispiel.

Der Gittervektor $\vec{v} \in \mathcal{L}(B)$ muss nach Definition eine ganzzahlige Linearkombination der orthogonalen Basisvektoren sein. Der Gittervektor $\vec{v}$ kann, wie bereits erwähnt, keine Linearkombination von mehreren Basisvektoren sein, also muss der Gittervektor $\vec{v}$ durch $\vec{v} = x_k\vec{b}_k$ darstellbar sein. Dabei ist $\vec{b}_k$ der Basisvektor, der zum Gittervektor $\vec{v}$ parallel ist. Damit der Gittervektor $\vec{v}$ kürzer als der Basisvektor $\vec{b}_k$ ist, muss $0 \leq x_k < 1$ gelten, was der Definition eines Gittervektors widerspricht, der eine ganzzahlige Linearkombination der Basisvektoren ist.

□

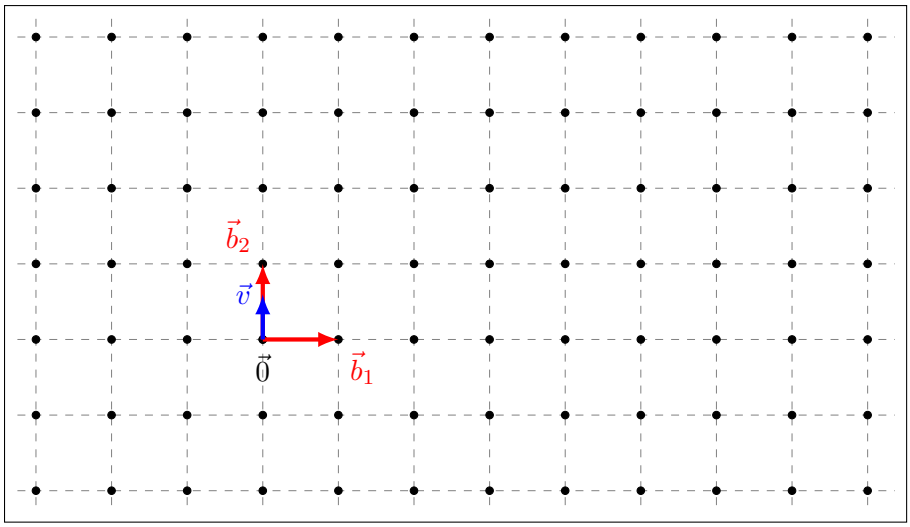


Abbildung 3.6: Es existieren in einer Orthogonalbasis keine kürzeren Gittervektoren als die Basisvektoren.

Ein Approximationsalgorithmus für SVP könnte aus diesem Grund einen minimalen Basisvektor als kürzesten Vektor im Gitter auswählen. Dies funktioniert bei paarweise orthogonalen Basisvektoren bzw. bei Basisvektoren, die paarweise „nahezu orthogonal“ sind und ein Hadamard-Verhältnis $\mathcal{H}(B)$ nahe an 1 besitzen, recht gut. Grundsätzlich lässt sich keine Aussage über die Güte eines mithilfe dieser Heuristik gefundenen „kürzesten Gittervektors“ machen, da die zur Approximation verwendete Basis auch aus langen nichtorthogonalen Basisvektoren bestehen kann.

Es wäre möglich, aus den langen und nichtorthogonalen Basisvektoren eine Basis, die aus kurzen und nahezu orthogonalen Basisvektoren besteht, zu berechnen. Dies wird als sogenannte *Gitterbasisreduktion* bezeichnet.

Es existieren mehrere verschiedene Begriffe der Reduziertheit einer Gitterbasis.

Definition 3.18 *Sei $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ die Basis des Gitters $\mathcal{L}$ und $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{Q}^m$ die zugehörige Gram-Schmidt Orthogonalbasis.*

Gilt für alle $i, j \in \{1, 2, \ldots, n\}$ und $j < i$ die Eigenschaft

$$\left| \frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} \right| \leq \frac{1}{2},$$

so wird die Basis $B = \left[\vec{b}_1 | \vec{b}_2 | \cdots | \vec{b}_n\right]$ als längenreduziert *bezeichnet.*

Die Eigenschaft $\left| \frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} \right| \leq \frac{1}{2}$ ist aus der „Berechnungssicht“ eines Algorithmus formuliert, der die Längenreduziertheit erreichen soll. Diese Eigenschaft kann wie folgt umformuliert werden, sodass ein Zusammenhang zwischen dem Basisvektor $\vec{b}_i$ und den Gram-Schmidt orthogonalisierten Basisvektoren $\vec{o}_j$ mit $i, j \in \{1, 2, \ldots, n\}$ und $j < i$ deutlich wird.

$$
\begin{aligned}
\left|\frac{\langle \vec{b}_i, \vec{o}_j\rangle}{\langle \vec{o}_j, \vec{o}_j\rangle}\right| &\leq \frac{1}{2} \\
\frac{|\langle \vec{b}_i, \vec{o}_j\rangle|}{\langle \vec{o}_j, \vec{o}_j\rangle} &\leq \frac{1}{2} \\
|\langle \vec{b}_i, \vec{o}_j\rangle| &\leq \frac{1}{2}\langle \vec{o}_j, \vec{o}_j\rangle \\
|\langle \vec{b}_i, \vec{o}_j\rangle| &\leq \frac{1}{2}\|\vec{o}_j\|^2 \\
\pm\langle \vec{b}_i, \vec{o}_j\rangle &\leq \frac{1}{2}\|\vec{o}_j\|^2 \\
\pm 2 \cdot \langle \vec{b}_i, \vec{o}_j\rangle &\leq \|\vec{o}_j\|^2 \\
\|\vec{b}_i\|^2 \pm 2 \cdot \langle \vec{b}_i, \vec{o}_j\rangle &\leq \|\vec{b}_i\|^2 + \|\vec{o}_j\|^2 \\
\|\vec{b}_i\|^2 &\leq \|\vec{b}_i\|^2 \pm 2 \cdot \langle \vec{b}_i, \vec{o}_j\rangle + \|\vec{o}_j\|^2 \\
\langle \vec{b}_i, \vec{b}_i\rangle &\leq \langle \vec{b}_i, \vec{b}_i\rangle \pm 2 \cdot \langle \vec{b}_i, \vec{o}_j\rangle + \langle \vec{o}_j, \vec{o}_j\rangle \\
\langle \vec{b}_i, \vec{b}_i\rangle &\leq \langle \vec{b}_i \pm \vec{o}_j, \vec{b}_i \pm \vec{o}_j\rangle \\
\sqrt{\langle \vec{b}_i, \vec{b}_i\rangle} &\leq \sqrt{\langle \vec{b}_i \pm \vec{o}_j, \vec{b}_i \pm \vec{o}_j\rangle} \\
\|\vec{b}_i\| &\leq \|\vec{b}_i \pm \vec{o}_j\|
\end{aligned}
$$

Gilt somit für die Basisvektoren eines Gitters und die zugehörige Gram-Schmidt orthogonalisierte Basis der Zusammenhang $\left|\frac{\langle \vec{b}_i, \vec{o}_j\rangle}{\langle \vec{o}_j, \vec{o}_j\rangle}\right| \leq \frac{1}{2}$ mit $i, j \in \{1, 2, \ldots, n\}$ und $j < i$, so gilt $\|\vec{b}_i\| \leq \|\vec{b}_i \pm \vec{o}_j\|$. Dies bedeutet, dass jeder Basisvektor $\vec{b}_i$ durch Addition bzw. Subtraktion der Gram-Schmidt orthogonalisierten Basisvektoren $\vec{o}_j$ mit $j < i$ nicht weiter verkürzt werden kann. Da $\vec{o}_k \leq \vec{b}_k$ für alle $k \in \{1, 2, \ldots, n\}$ gilt, ist dies nur eine Approximation einer „guten“ (kurzen und nahezu orthogonalen) Basis.

Der Algorithmus 3.3 mit Laufzeit $\mathcal{O}(n^2)$ erzeugt aus jeder gegebenen Gitterbasis eine längenreduzierte Gitterbasis.

Der Schleifenindex j in Zeile 2 muss rückwärts laufen, da $\text{span}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_j) = \text{span}(\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_j)$, aber $\text{span}(\vec{b}_j, \vec{b}_{j+1}, \ldots, \vec{b}_n)$ ist nicht zwingend gleich $\text{span}(\vec{o}_j, \vec{o}_{j+1}, \ldots, \vec{o}_n)$. Die Gram-Schmidt Orthogonalbasis wird durch diesen Algorithmus nicht verändert.

Im Folgenden wird kurz die Korrektheit von Algorithmus 3.3 gezeigt.

Algorithmus 3.3 : Längenreduktion einer Basis

Eingabe : Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ für ein Gitter $\mathcal{L}$ und $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{Q}^m$ die zugehörige Gram-Schmidt orthogonalisierte Basis

Ausgabe : Basisvektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n \in \mathbb{Q}^m$ für das Gitter $\mathcal{L}$, sodass $\left|\frac{\langle \vec{b}'_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| \leq \frac{1}{2}$ für alle Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ mit $i, j \in \{1, 2, \ldots, n\}$ und $i \neq j$ gilt

```
1 for i = 2, 3, ..., n do
2 |   for j = (i − 1), (i − 2), ..., 1 do
3 |   |   b_i ← b_i − ⌊ ⟨b_i, o_j⟩ / ⟨o_j, o_j⟩ ⌉ · b_j
4 |   end for
5 end for
```

Proposition 3.19 *Seien $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_i \in \mathbb{Q}^m$ die i ersten Basisvektoren eines Gitters und $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_i \in \mathbb{Q}^m$ die i zugehörigen Gram-Schmidt Orthogonalbasisvektoren, so gilt:*

$$\frac{\langle \vec{b}_i, \vec{o}_i \rangle}{\langle \vec{o}_i, \vec{o}_i \rangle} = 1\,.$$

Beweis:

Die Berechnungsformel des Gram-Schmidt Orthogonalisierungsverfahren kann wie folgt umgeschrieben werden:

$$\vec{o}_i = \vec{b}_i - \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)$$

$$\vec{b}_i = \vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i)\,.$$

Daraus folgt:

$$\frac{\langle \vec{b}_i, \vec{o}_i \rangle}{\langle \vec{o}_i, \vec{o}_i \rangle} = \frac{\left\langle \vec{o}_i + \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_i \right\rangle}{\langle \vec{o}_i, \vec{o}_i \rangle} = \frac{\langle \vec{o}_i, \vec{o}_i \rangle + \left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_i \right\rangle}{\langle \vec{o}_i, \vec{o}_i \rangle}$$

Da die Vektoren $\vec{o}_i$ und $\vec{o}_j$ für $i, j \in \{1, 2, \ldots, n\}$ und $j < i$ paarweise orthogonal zueinander sind, gilt auch $\left\langle \sum_{j=1}^{i-1} p_{\vec{o}_j}(\vec{b}_i), \vec{o}_i \right\rangle = 0$. Daraus folgt weiter:

$$= \frac{\langle \vec{o}_i, \vec{o}_i \rangle}{\langle \vec{o}_i, \vec{o}_i \rangle} = 1\,.$$

□

Satz 3.20 *Sei $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ eine beliebige Basis des Gitters $\mathcal{L}$ die Eingabe für den Algorithmus 3.3, so sind die ausgegebenen Basisvektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n \in \mathbb{Q}^m$ längernreduziert.*

Beweis:

Als Erstes wird gezeigt, dass durch Algorithmus 3.3 nur Gittervektoren entstehen können. Durch die Zuweisung in Zeile 3 wird von einem Gittervektor das ganzzahlige Vielfache eines anderen Gittervektors addiert bzw. subtrahiert. Diese neu berechneten Vektoren sind nach Definition eines Gitters wiederum Gittervektoren. Des Weiteren ist diese Zuweisung in den drei Operationen von Folgerung 2.14 enthalten, die eine Transformation der Basisvektoren angeben, sodass diese transformierten Basisvektoren ein äquivalentes Gitter erzeugen.

Als Letztes wird gezeigt, dass die durch Algorithmus 3.3 berechneten Basisvektoren längenreduziert sind, also $\left|\frac{\langle \vec{b}'_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| \leq \frac{1}{2}$ gilt, wobei $\vec{b}'_i$ der i. vom Algorithmus ausgegebene Basisvektor ist. Im Folgenden gilt $\mu_{i,j} =_{\text{def}} \frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}$.

$$\left|\frac{\langle \vec{b}'_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| = \left|\frac{\langle \vec{b}_i - \lfloor \mu_{i,j} \rceil \, \vec{b}_j, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right|$$
$$= \left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} - \frac{\langle \lfloor \mu_{i,j} \rceil \, \vec{b}_j, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| = \left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} - \lfloor \mu_{i,j} \rceil \, \frac{\langle \vec{b}_j, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| .$$

Der Term $\frac{\langle \vec{b}_j, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}$ ist nach Proposition 3.19 gleich Eins. Daraus folgt:

$$= \left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} - \lfloor \mu_{i,j} \rceil \cdot 1\right| = \left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} - \lfloor \mu_{i,j} \rceil\right| .$$

Zur Verdeutlichung wird anstatt $\mu_{i,j}$ wieder $\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}$ eingesetzt:

$$= \left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} - \left\lfloor \frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} \right\rceil\right| \leq \frac{1}{2} .$$

□

Eine weitere Eigenschaft für die Reduziertheit einer Basis ist die sogenannte paarweise reduzierte Basis.

Definition 3.21 *Gilt für alle Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ eines Gitters mit $i, j \in \{1, 2, \ldots, n\}$ und $i \neq j$ die Eigenschaften:*

- $\|\vec{b}_1\| \leq \|\vec{b}_2\| \leq \cdots \leq \|\vec{b}_n\|$
- $\left|\frac{\langle \vec{b}_i, \vec{b}_j \rangle}{\langle \vec{b}_j, \vec{b}_j \rangle}\right| \leq \frac{1}{2}$

so wird die Basis auch paarweise reduziert *genannt. Dies bedeutet, dass durch Addition bzw. Subtraktion zweier Basisvektoren keine kürzeren Basisvektoren erzeugt werden können.*

Satz 3.22 *Aus $\left|\frac{\langle \vec{b}_i, \vec{b}_j \rangle}{\langle \vec{b}_j, \vec{b}_j \rangle}\right| \leq \frac{1}{2}$ folgen die beiden Eigenschaften:*

- $\|\vec{b}_i\| \leq \|\vec{b}_i + \vec{b}_j\|$
- $\|\vec{b}_i\| \leq \|\vec{b}_i - \vec{b}_j\|$.

Beweis:

Beweis analog zur Umformung der Eigenschaft der Längenreduziertheit. □

Algorithmus 3.4 erzeugt nach endlich vielen Schritten für jede Basis eine paarweise reduzierte Basis. Die Laufzeit des Algorithmus ist nicht polynomiell [Sch08, S. 23].

Algorithmus 3.4 : Paarweise Reduktion einer Basis (aus [Sch08, S. 22]).

Eingabe : Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ für ein Gitter $\mathcal{L}$.
Ausgabe : Basisvektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n \in \mathbb{Q}^m$ für das Gitter $\mathcal{L}$, sodass $\frac{|\langle \vec{b}_i, \vec{b}_j \rangle|}{\langle \vec{b}_j, \vec{b}_j \rangle} \leq \frac{1}{2}$ für alle Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ mit $i, j \in \{1, 2, \ldots, n\}$ und $i \neq j$, sowie $\|\vec{b}_1\| \leq \|\vec{b}_2\| \leq \cdots \leq \|\vec{b}_n\|$ gilt.

```
1  Ordne b_1, b_2, ..., b_n, sodass ||b_1|| ≤ ||b_2|| ≤ ··· ≤ ||b_n|| gilt
2  for i = 1, 2, ..., n do
3  |  for j = 1, 2, ..., (i − 1) do
4  |  |  r ← ⟨b_i, b_j⟩ / ⟨b_j, b_j⟩
5  |  |  if |r| > 1/2 then
6  |  |  |  b_i ← b_i − ⌊r⌉ b_j
7  |  |  |  goto 1
8  |  |  end if
9  |  end for
10 end for
```

Eine paarweise reduzierte Basis ist nicht maximal reduziert, wie folgendes Beispiel zeigt.

Beispiel: (aus [Sch08, S. 23])

Die Basisvektoren $\vec{b}_1, \vec{b}_2, \vec{b}_3$ seien die Basisvektoren des Gitters $\mathcal{L}(\vec{b}_1, \vec{b}_2, \vec{b}_3)$ mit Rang 3. Dabei sind die Basisvekotren wie folgt gegeben:

$$\vec{b}_1 =_{\text{def}} \begin{pmatrix} 9 \\ 0 \\ 0 \end{pmatrix}, \ \vec{b}_2 =_{\text{def}} \begin{pmatrix} -3 \\ 8 \\ 0 \end{pmatrix}, \ \vec{b}_3 =_{\text{def}} \begin{pmatrix} -3 \\ -5 \\ 6 \end{pmatrix}.$$

Für die Länge dieser Basisvektoren gilt:

$$\|\vec{b}_1\| = \sqrt{81}, \ \|\vec{b}_2\| = \sqrt{73}, \ \|\vec{b}_3\| = \sqrt{70}.$$

Dies ist eine paarweise reduzierte Basis, das bedeutet, dass durch paarweise Addition bzw. Subtraktion der Basisvektoren kein kürzerer Basisvektor entsteht:

$$\vec{b}_1 + \vec{b}_2 = \begin{pmatrix} 6 \\ 8 \\ 0 \end{pmatrix}, \ \vec{b}_1 + \vec{b}_3 = \begin{pmatrix} 6 \\ -5 \\ 6 \end{pmatrix}, \ \vec{b}_2 + \vec{b}_3 = \begin{pmatrix} -6 \\ 3 \\ 6 \end{pmatrix}$$

$$\vec{b}_1 - \vec{b}_2 = \begin{pmatrix} 12 \\ -8 \\ 0 \end{pmatrix}, \ \vec{b}_1 - \vec{b}_3 = \begin{pmatrix} 12 \\ 5 \\ -6 \end{pmatrix}, \ \vec{b}_2 - \vec{b}_3 = \begin{pmatrix} 0 \\ 13 \\ -6 \end{pmatrix}.$$

Für die Länge dieser Gittervektoren gilt:

$$\|\vec{b}_1 + \vec{b}_2\| = \sqrt{100}, \ \|\vec{b}_1 + \vec{b}_3\| = \sqrt{97}, \ \|\vec{b}_2 + \vec{b}_3\| = \sqrt{81}$$

$$\|\vec{b}_1 - \vec{b}_2\| = \sqrt{208}, \ \|\vec{b}_1 - \vec{b}_3\| = \sqrt{205}, \ \|\vec{b}_2 - \vec{b}_3\| = \sqrt{205}.$$

Diese Gittervektoren sind Basisvektoren für das Gitter, da diese nach der Regel $\vec{b}_i \leftarrow \vec{b}_i + k\vec{b}_j$, wobei $k \in \mathbb{Z}$ ist (siehe Folgerung 2.14) entstanden sind.

Trotzdem ist die Basis nicht maximal reduziert, da mindestens ein kürzerer Basisvektor existiert:

$$(\vec{b}_1 + \vec{b}_2) + \vec{b}_3 = \begin{pmatrix} 3 \\ 3 \\ 6 \end{pmatrix} \text{ und } \|(\vec{b}_1 + \vec{b}_2) + \vec{b}_3\| = \sqrt{54}.$$

Dieser Gittervektor ist demnach kürzer als die drei Basisvektoren, die paarweise reduziert sind. Des Weiteren ist es ein Basisvektor, da dieser wiederum mithilfe der Regel $\vec{b}_i \leftarrow \vec{b}_i + k\vec{b}_j$, wobei $k \in \mathbb{Z}$ (siehe Folgerung 2.14) entstanden ist. Diese Regel muss hierfür zweimal angewendet werden, was durch die Klammerung hervorgehoben wurde.

Bis Anfang der 1980er Jahre war kein Polynomialzeitslgorithmus bekannt, der eine Gitterbasisreduktion durchführen konnte, sodass in diesem reduzierten Gitter das „Shortest Vec-

tor Problem“ bis auf einen bestimmten Ungenauigkeitsfaktor approximiert werden kann. Es wird mittlerweile angenommen, dass kein Polyomialzeitalgorithmus für das „Shortest Vector Problem“ existiert, dessen approximierte Lösung maximal um einen polynomiellen Faktor schlechter ist, als die exakte Lösung des Problems [BB09, S. 149]. Im Jahr 1982 wurde der LLL-Approximationsalgorithmus vorgestellt, der eine Gitterbasis so reduzieren kann, dass das „Shortest Vector Problem“ mithilfe der reduzierten Basis bis auf einen exponentiellen Ungenauigkeitsfaktor approximiert werden kann. Der LLL-Algorithmus ist einer der bekanntesten und am meisten untersuchte Algorithmus für Gitterprobleme [BB09, S. 148]. Obwohl der LLL-Algorithmus bspw. durch Schnorr in [Sch87] verallgemeinert wurde, wird hier nur der „klassische“ LLL-Algorithmus vorgestellt.

Der LLL-Algorithmus erzeugt eine Basis, die LLL-reduziert ist. Dieser Begriff ist wie folgt definiert:

Definition 3.23 *Seien* $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ *die Basisvektoren des Gitters* $\mathcal{L}$ *und* $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{Q}^m$ *die zugehörige Gram-Schmidt Orthogonalbasis.*

Gilt für alle $i, j \in \{1, 2, \ldots, n\}$ *und* $j < i$ *die Eigenschaften:*

- $\left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| \leq \frac{1}{2}$ (Längenreduziertheit)
- $\left\|\vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\vec{o}_{i-1}\right\|^2 \geq \frac{3}{4}\|\vec{o}_{i-1}\|^2$ (Lovász-Eigenschaft)

so wird die Basis LLL-reduziert *genannt.*

Proposition 3.24 *Seien* $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ *die Basisvektoren des Gitters* $\mathcal{L}$ *und* $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{Q}^m$ *die zugehörige Gram-Schmidt Orthogonalbasis.*
Gilt $\left\|\vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\vec{o}_{i-1}\right\|^2 \geq \frac{3}{4}\|\vec{o}_{i-1}\|^2$*, so gilt auch:*

$$\|\vec{o}_i\|^2 \geq \left(\frac{3}{4} - \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\right)^2\right)\|\vec{o}_{i-1}\|^2 .$$

Beweis:

$$\left\|\vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\vec{o}_{i-1}\right\|^2 \geq \frac{3}{4}\|\vec{o}_{i-1}\|^2$$

$$\left\langle \vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\vec{o}_{i-1}, \vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\vec{o}_{i-1} \right\rangle \geq \frac{3}{4}\|\vec{o}_{i-1}\|^2$$

$$\langle \vec{o}_i, \vec{o}_i \rangle + 2 \left\langle \vec{o}_i, \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} \right\rangle + \left\langle \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1}, \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} \right\rangle \geq \frac{3}{4} \|\vec{o}_{i-1}\|^2$$

Es gilt $2 \left\langle \vec{o}_i, \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} \right\rangle = 0$, da die Vektoren $\vec{o}_i$ und $\vec{o}_{i-1}$ orthogonal zueinander sind.

$$\begin{aligned}
\langle \vec{o}_i, \vec{o}_i \rangle + \left\langle \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1}, \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} \right\rangle &\geq \frac{3}{4} \|\vec{o}_{i-1}\|^2 \\
\langle \vec{o}_i, \vec{o}_i \rangle + \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \right)^2 \langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle &\geq \frac{3}{4} \|\vec{o}_{i-1}\|^2 \\
\|\vec{o}_i\|^2 + \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \right)^2 \|\vec{o}_{i-1}\|^2 &\geq \frac{3}{4} \|\vec{o}_{i-1}\|^2 \\
\|\vec{o}_i\|^2 &\geq \frac{3}{4} \|\vec{o}_{i-1}\|^2 - \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \right)^2 \|\vec{o}_{i-1}\|^2 \\
\|\vec{o}_i\|^2 &\geq \left(\frac{3}{4} - \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \right)^2 \right) \|\vec{o}_{i-1}\|^2
\end{aligned}$$

□

Proposition 3.24 zeigt, dass die Gram-Schmidt Orthogonalvektoren einer LLL-reduzierten Basis nicht beliebig kurz werden dürfen. Es existiert also eine Mindestlänge, die die Gram-Schmidt Orthogonalvektoren einhalten müssen, in Bezug auf den jeweiligen vorherigen Gram-Schmidt Orthogonalvektor.

Hilfssatz 3.25 *Sei $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ eine LLL-reduzierte Basis eines Gitters und $\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_n \in \mathbb{Q}^m$ die zugehörige Gram-Schmidt Orthogonalbasis, dann gilt stets:*

$$\|\vec{o}_i\|^2 \geq \frac{1}{2} \|\vec{o}_{i-1}\|^2 .$$

Beweis:

Da $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ eine LLL-reduzierte Basis ist, gilt nach der Lovász-Eigenschaft folgender Zusammenhang zwischen den Längen der beiden Gram-Schmidt Orthogonalvektoren $\vec{o}_i$ und $\vec{o}_{i-1}$:

$$\|\vec{o}_i\|^2 \geq \left(\frac{3}{4} - \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \right)^2 \right) \|\vec{o}_{i-1}\|^2 .$$

Da nach der Längenreduziertheit $0 < \left|\frac{\langle \vec{b}_i, \vec{o}_{i-1}\rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1}\rangle}\right| \leq \frac{1}{2}$ gilt, folgt daraus:

$$0 < \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1}\rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1}\rangle}\right)^2 \leq \frac{1}{4}\,.$$

Die obere Grenze wird in die Formel der Lovász-Eigenschaft eingesetzt:

$$\|\vec{o}_i\|^2 \geq \left(\frac{3}{4} - \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1}\rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1}\rangle}\right)^2\right)\|\vec{o}_{i-1}\|^2 \geq \left(\frac{3}{4} - \frac{1}{4}\right)\|\vec{o}_{i-1}\|^2 = \frac{1}{2}\|\vec{o}_{i-1}\|^2$$

□

Hilfssatz 3.25 kann benutzt werden, um folgenden Satz zu beweisen, der zeigt, wofür eine LLL-reduzierte Gitterbasis verwendet werden kann.

Satz 3.26 *Sei $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ eine LLL-reduzierte Basis, dann gilt:*

$$\|\vec{b}_1\| \leq \sqrt{2^{n-1}}\lambda_1(\mathcal{L})\,,$$

wobei n dem Rang des Gitters entspricht.

Beweis:

Aus Hilfssatz 3.25 folgt:

$$\|\vec{o}_n\|^2 \geq \frac{1}{2}\|\vec{o}_{n-1}\|^2 \geq \cdots \geq \left(\frac{1}{2}\right)^{n-1}\|\vec{o}_1\|^2 = \left(\frac{1}{2}\right)^{n-1}\|\vec{b}_1\|^2\,.$$

Die letzte Gleichung folgt, da bei der Gram-Schmidt Orthogonalisierung der erste Basisvektor als Gram-Schmidt Orthogonalvektor übernommen wird und somit $\vec{o}_1 = \vec{b}_1$ gilt. Für den k. Gram-Schmidt Orthogonalvektor gilt nach Hilfssatz 2.35 $0 < \min_k \|\vec{o}_k\| \leq \lambda_1(\mathcal{L})$. Daraus folgt:

$$\|\vec{b}_1\|^2 \leq \frac{1}{\left(\frac{1}{2}\right)^{k-1}}\|\vec{o}_k\|^2 \leq \frac{1}{\left(\frac{1}{2}\right)^{n-1}}\|\vec{o}_k\|^2 = \frac{1}{2^{-(n-1)}}\|\vec{o}_k\|^2 = 2^{(n-1)}\|\vec{o}_k\|^2$$

$$\|\vec{b}_1\| \leq \sqrt{2^{(n-1)}}\|\vec{o}_k\| \leq \sqrt{2^{(n-1)}}\min_k\|\vec{o}_k\| \leq \sqrt{2^{(n-1)}}\lambda_1(\mathcal{L}).$$

□

Der LLL-Algorithmus erzeugt in polynomieller Laufzeit aus einer beliebigen Gitterbasis eine LLL-reduzierte Basis eines äquivalenten Gitters. Als Approximation eines kürzesten

Vektors im Gitter wird der Basisvektor $\vec{b}_1$ der LLL-reduzierten Basis verwendet. Satz 3.26 zeigt, dass in einer LLL-reduzierten Basis ein kürzester Gittervektor approximiert werden kann, dessen Güte (in diesem Fall die Länge des Vektors) maximal um einen exponentiellen Faktor schlechter ist, als einer der tatsächlich kürzesten Gittervektoren, deren Länge dem sukzessiven Minimum λ_1 entspricht. Der LLL-Algorithmus ist demnach ein Algorithmus für $\mathsf{SVP}_{\sqrt{2^{(n-1)}}}$.

Algorithmus 3.5 zeigt den LLL-Algorithmus, der aus einer beliebigen Basis eines Gitters eine LLL-reduzierte Basis berechnen kann.

Algorithmus 3.5 : LLL-Gitterreduktionsalgorithmus

Eingabe : Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n \in \mathbb{Q}^m$ für ein Gitter $\mathcal{L}$.
Ausgabe : LLL-reduzierte Basisvektoren $\vec{b}'_1, \vec{b}'_2, \ldots, \vec{b}'_n \in \mathbb{Q}^m$ für das Gitter $\mathcal{L}$, sodass $\left|\frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle}\right| \leq \frac{1}{2}$ und $\|\vec{o}_i\|^2 \geq \left(\frac{3}{4} - \left(\frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle}\right)^2\right) \|\vec{o}_{i-1}\|^2$ für alle Basisvektoren $\vec{b}_i$ und $\vec{b}_j$ mit $i, j \in \{1, 2, \ldots, n\}$ und $i \neq j$ gilt.

1 $i \leftarrow 2$
2 $\vec{o}_1 \leftarrow \vec{b}_1$
3 **while** $i \leq n$ **do**
4 **for** $j = (i-1), (i-2), \ldots, 1$ **do**
5 $\vec{b}_i = \vec{b}_i - \left\lfloor \frac{\langle \vec{b}_i, \vec{o}_j \rangle}{\langle \vec{o}_j, \vec{o}_j \rangle} \right\rceil \cdot \vec{b}_j$
6 **end for**
7 **if** $\left\| \vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} \right\|^2 \geq \frac{3}{4} \|\vec{o}_{i-1}\|^2$ **then**
8 $i \leftarrow i + 1$
9 **else**
10 *tausche* $\vec{b}_{i-1}$ *mit* $\vec{b}_i$
11 $i \leftarrow \max(i-1, 2)$
12 **end if**
13 **end while**

Anmerkung: Bei der Einführung des Gram-Schmidt Orthogonalisierungsverfahren wurde erläutert, dass die erzeugte Gram-Schmidt Orthogonalbasis von der Reihenfolge der eingegebenen Basisvektoren abhängig ist. In Algorithmus 3.5 wird in Zeile 10 die Reihenfolge der Basisvektoren vertauscht. Die notwendige Aktualisierung der Gram-Schmidt Orthogonalbasis nach der Vertauschung in Zeile 10 wird in der obigen Darstellung des LLL-Algorithmus vernachlässigt. Eine ausführlich kommentierte Implementierung des LLL-Algorithmus in Mathematica findet sich im Anhang A.

Im Folgenden werden die einzelnen Schritte dieses Algorithmus für eine Iteration der äußeren Schleife erläutert. Eine ausführliche Analyse der Korrektheit und der Laufzeit des

Algorithmus findet sich in [LLL82].

Als Erstes wird erläutert, dass die vom LLL-Algorithmus berechnete Gitterbasis ein äquivalentes Gitter erzeugt. Die einzigen Anweisungen im Algorithmus, welche die Basisvektoren verändern, sind in Zeile 5 und Zeile 10 enthalten. Dass die Anweisung in Zeile 5 nur Gittervektoren ergeben und diese auch ein äquivalentes Gitter erzeugen, wurde bereits bei der Erläuterung von Algorithmus 3.3 angegeben. Das Vertauschen in Zeile 10 erhält die Eigenschaft, dass die Vektoren Gittervektoren sind. Des Weiteren ist die Vertauschung zweier Basisvektoren in den drei Operationen von Folgerung 2.14 enthalten und diese Vertauschung verändert das durch die Basisvektoren erzeugte Gitter nicht.

In der i. Iteration der äußeren Schleife in Zeile 3 ist garantiert, dass die Basis bestehend aus den Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{i-1}$ LLL-reduziert ist. Die innere Schleife von Zeile 4 bis Zeile 6 sorgt dafür, dass die Längenreduziertheit für die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_i$ und die zugehörige Gram-Schmidt Orthogonalbasis erreicht wird.

In Zeile 7 wird überprüft, ob die Gram-Schmidt Orthogonalvektoren $\vec{o}_{i-1}$ und $\vec{o}_i$ die Lovász-Eigenschaft erfüllen. Ist dies der Fall, so wird der Laufindex i der äußeren Schleife inkrementiert. Sofern die Lovász-Eigenschaft nicht erfüllt ist, werden die beiden Basisvektoren $\vec{b}_{i-1}$ und $\vec{b}_i$ vertauscht. Der Effekt dieser Vertauschung wird im Folgenden verdeutlicht.

Werden die Befehle in den Zeilen 9 bis Zeile 11 abgearbeitet, so muss das Gegenteil der Bedingung in Zeile 7 gelten. Demnach gilt in diesen Zeilen:

$$\left\| \vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} \right\|^2 < \frac{3}{4} \|\vec{o}_{i-1}\|^2 .$$

Die Vertauschung der beiden Basisvektoren in Zeile 10 hat u. a. die folgende Auswirkung auf den Gram-Schmidt Orthogonalvektor $\vec{o}_{i-1}$:

$$\vec{o}'_{i-1} \leftarrow \vec{o}_i + \frac{\langle \vec{b}_i, \vec{o}_{i-1} \rangle}{\langle \vec{o}_{i-1}, \vec{o}_{i-1} \rangle} \vec{o}_{i-1} .$$

Zusammen bedeuten diese beiden Eigenschaften, dass für den neuen Gram-Schmidt Ortohgonalvektor $\vec{o}'_{i-1}$, nach dem Vertauschungsschritt, folgende Gleichung gilt:

$$\|\vec{o}'_{i-1}\|^2 < \frac{3}{4} \|\vec{o}_{i-1}\|^2 .$$

Da $\|\vec{o}'_{i-1}\| > 0$ ist, folgt daraus, dass die Länge des Gram-Schmidt Orthogonalvektor durch die Vertauschung der Basisvektoren echt verkürzt wurde. Die Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{i-1}$ sind nach der Vertauschung nicht zwingend LLL-reduziert. Aus diesem Grund wird der Schleifenindex i auf den Wert $i-1$ gesetzt bzw. auf 2, falls $i = 2$ gilt. Nach dieser Weise werden Schritt für Schritt die Gram-Schmidt Orthogonalvektoren verkürzt und nach der

Terminierung des LLL-Algorithmus ist die ausgegebene Gitterbasis LLL-reduziert.

Anmerkung: Es wird kurz hervorgehoben, dass im Vertauschungsschritt in Zeile 10 auch der Basisvektor $\vec{b}_1$ mit dem Basisvektor $\vec{b}_2$ vertauscht werden kann. Dies bedeutet, dass für den Gram-Schmidt Orthogonalvektor $\vec{o}_1' \leftarrow \vec{b}_2$ gilt. Dieser Spezialfall muss in einer Implementierung des Algorithmus berücksichtigt werden.

Die Approximation eines kürzesten Vektors mithilfe des LLL-Algorithmus ist in einem Gitter mit einem kleinen Rang n gut, in höheren Dimension ($500 \leq n \leq 1000$) meist schlecht [HPS08, S. 403 und 418].

Die beiden hier vorgestellten Probleme, das „Closest Vector Problem“ und das „Shortest Vector Problem“ können unter bestimmten Voraussetzungen in Verbindung gesetzt werden. Das „Closest Vector Problem“ mit Dimension m kann auf ein „Shortest Vector Problem“ mit Dimension $m + 1$ reduziert werden [Gal12, S. 373]. Das „Closest Vector Problem“ kann für LLL-reduzierte Gitterbasen auch, wie bereits erwähnt, mithilfe der vorgestellten Heuristiken von Babai gelöst werden. Eine Approximation für das „Closest Vector Problem“ kann den LLL-Algorithmus auf eine beliebige eingegebene Gitterbasis anwenden und in der ausgegebenen LLL-reduzierten Gitterbasis eine der beiden Heuristiken von Babai anwenden.

Kapitel 4

Analyse der Berechnungsprobleme

Dieses Kapitel verdeutlicht, dass die im letzten Kapitel eingeführten Gitterprobleme „schwierig“ sind. Dies bedeutet, dass für die exakte Lösung, als auch für eine approximierte Lösung, die höchstens um einen polynomiellen Faktor schlechter als die exakte Lösung ist, kein Polynomialzeitalgorithmus existiert.

Zu Beginn werden die benötigten Grundlagen der Komplexitätstheorie wiederholt, sodass bspw. die NP-Härte des „Closest Vector Problem“ gezeigt werden kann.

Hinweis: Im Folgenden wird das „Closest Vector Problem“ mit CVP und das „Shortest Vector Problem“ mit SVP abgekürzt.

4.1 Komplexitätstheoretische Grundlagen

Dieser Abschnitt erläutert kurz die komplexitätstheoretischen Grundlagen, die im Folgenden benötigt werden. Eine Einführung in die Komplexitätstheorie, sowie Komplexitätsklassen, wie P, NP und deren Eigenschaften gibt bspw. [Sip06].

Hinweis: In diesem Abschnitt werden die Begriffe NP-hart und Instanz (eines Problems) verwendet, obwohl diese Begriffe historisch falsch aus der englischen Sprache übersetzt wurden.

Definition 4.1 *(nach [Sip06, S. 272])*

Eine Sprache A ist Karp-reduzierbar *auf eine Sprache B, wenn eine berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ existiert, sodass folgende drei Eigenschaften für f gelten:*

Für alle $w \in \Sigma^*$ berechnet die Funktion f den Funktionswert $f(w)$ in polynomieller Zeit.

$$w \in A \to f(w) \in B\,.$$

$$w \notin A \rightarrow f(w) \notin B\,.$$

Dabei heißt f die Karp-Reduktion *von Sprache A nach Sprache B. Dieser Zusammenhang zwischen den Sprachen A und B wird mithilfe der folgenden Schreibweise ausgedrückt:*

$$A \leq_{\mathrm{p}} B\,.$$

Die Abbildung 4.1 zeigt eine Karp-Reduktion zwischen zwei Sprachen A und B.

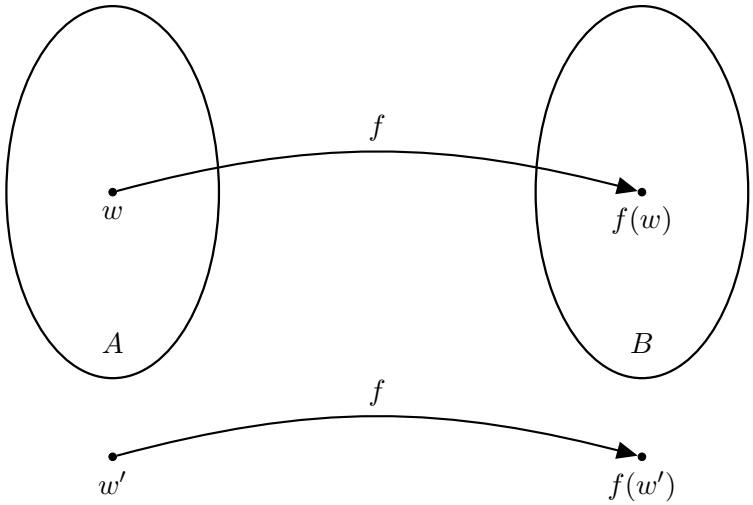


Abbildung 4.1: Eine Karp-Reduktion von Sprache A nach Sprache B (nach [Sip06, S. 207]).

Karp-Reduktionen werden u. a. benötigt, um beweisen zu können, dass eine Sprache NP-hart bzw. NP-vollständig ist.

Definition 4.2 *Sei A eine Sprache. Sind alle Sprachen B in NP Karp-reduzierbar auf A, so wird A als* NP-hart *bezeichnet.*

Der Begriff der NP-Härte kann wie folgt erweitert werden:

Definition 4.3 *Sei A eine NP-harte Sprache. Gilt A in NP, so wird A als* NP-vollständig *bezeichnet.*

Es gilt, dass alle NP-vollständigen Probleme auch NP-hart sind. Um zu zeigen, dass eine Sprache A eine NP-harte Sprache ist, existieren mehrere Möglichkeiten: Es muss gezeigt werden, dass alle anderen Sprachen B in NP Karp-reduzierbar auf die Sprache A sind. Analog bedeutet dies, dass gezeigt werden muss, dass ein NP-vollständiges Problem C auf die Sprache A Karp-reduziert werden kann.

Im Folgenden wird anstatt von einer Sprache A immer von einem Problem A gesprochen. Anstatt von einem Wort w einer Sprache A wird analog von einer Instanz w eines Problems A gesprochen.

Ein Problem, was im Folgenden benötigt wird, ist das sogenannte „Subset-Sum Problem“:

Definition 4.4 *Das „Subset-Sum Problem (SUBSET SUM)“ ist wie folgt definiert:*

PROBLEM: *SUBSET SUM*

EINGABE: *Eine endliche Menge* $A =_{\text{def}} \{a_1, a_2, \ldots, a_n\}$ *mit* $a_i \in \mathbb{Q}$ *für* $i \in \{1, 2, \ldots, n\}$, $n \in \mathbb{N}$ *und* $T \in \mathbb{Q}$.

FRAGE: *Existiert eine endliche Teilmenge* $B \subseteq A =_{\text{def}} \{b_1, b_2, \ldots, b_m\}$ *mit* $m \in \mathbb{N}$, *sodass* $m \leq n$ *und* $\sum_{i=1}^{m} b_i = T$ *gilt?*

Alternativ kann das Problem auch wie folgt definiert werden:

PROBLEM: *SUBSET SUM**

EINGABE: *Eine endliche Menge* $A =_{\text{def}} \{a_1, a_2, \ldots, a_n\}$ *mit* $a_i \in \mathbb{Q}$ *für* $i \in \{1, 2, \ldots, n\}$, $n \in \mathbb{N}$ *und* $T \in \mathbb{Q}$.

FRAGE: *Existiert eine endliche Menge* $X =_{\text{def}} \{x_1, x_2, \ldots, x_n\}$ *mit* $x_i \in \{0, 1\}$, *sodass* $\sum_{i=1}^{n} x_i a_i = T$ *gilt? Die Elemente in der Menge* X *entscheiden, welches Element aus der Menge* A *zur Konstruktion der Summe* T *benutzt wird und welches nicht.*

Hilfssatz 4.5 *Das „Subset-Sum Problem (SUBSET SUM)“ ist NP-vollständig.*

Beweis: siehe [GJ79, S. 223] oder [Sip06, S. 292 ff.].

Im Folgenden wird eine Instanz von SUBSET SUM bzw. SUBSET SUM* verkürzt mithilfe eines $n+1$-Tupels $(a_1, a_2, \ldots, a_n, T)$ angegeben.

4.2 Komplexität von CVP

Als Erstes wird CVP als Entscheidungsproblem formuliert:

Definition 4.6 *Das „Decision Closest Vector Problem (CVP$_D$)“ ist wie folgt definiert:*

PROBLEM: *CVP$_D$*

EINGABE: *Eine Basis* $B \in \mathbb{Q}^{m \times n}$ *mit* $m \geq n$ *eines Gitters* $\mathcal{L}$, *ein Vektor* $\vec{q} \in \operatorname{span}(B)$ *und ein* $r \in \mathbb{Q}$.

FRAGE: *Existiert ein Gittervektor* $\vec{v} \in \mathcal{L}$, *sodass gilt:*

$$\|\vec{q} - \vec{v}\| \leq r\,.$$

Satz 4.7 *Es gilt SUBSET SUM $\leq_{\mathrm{p}}$ CVP und somit ist CVP ein NP-hartes Problem.*

Beweis: (nach [MG02, S. 48 f.])

Für die Reduktion wird folgendes Gitter verwendet:

$$B =_{\mathrm{def}} \begin{pmatrix} 2 & 0 & \cdots & 0 \\ 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2 \\ a_1 & a_2 & \cdots & a_n \end{pmatrix} \in \mathbb{Q}^{(n+1)\times n},\ \vec{q} =_{\mathrm{def}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ T \end{pmatrix} \in \mathbb{Q}^{n+1},\ r =_{\mathrm{def}} \sqrt{n}\,,$$

wobei $n+1$ der Dimension und n dem Rang des Gitters entspricht. Im Folgenden wird dieses konstruierte Gitter verkürzt als Funktion f bezeichnet. Die Konstruktion eines solchen Gitters aus einer Instanz des SUBSET SUM-Problems kann in polyomieller Laufzeit durchgeführt werden.

In dem vorher definierten Entscheidungsproblem wird gefordert, dass $r \in \mathbb{Q}$ ist. Dies ist bei $r = \sqrt{n}$ nicht garantiert. Sollte $\sqrt{n}$ eine irrationale Zahl sein, so kann diese durch eine beliebige rationale Approximation im Intervall $[\sqrt{n}, \sqrt{n+1})$ ersetzt werden (siehe [RT33, S. 89 ff.] und [MG02, S. 49]).

Zuerst wird gezeigt, dass für eine beliebige Instanz $w \in$ SUBSET SUM $\rightarrow f(w) \in$ CVP gilt. Sei $(a_1, a_2, \ldots, a_n, T)$ eine Instanz des SUBSET SUM-Problems und $\vec{x} = (x_1, x_2, \ldots, x_n)^T$ die zugehörige Lösung dieser Instanz, sodass $\sum_{i=1}^{n} x_i a_i = T$ mit $x_i \in \{0, 1\}$ gilt. Der aus dieser Instanz konstruierte Gittervektor ist $\vec{v} = B\vec{x} = (2x_1, 2x_2, \ldots, 2x_n, \sum_{i=1}^{n} x_i a_i)^T$. Für die Distanz dieses Gittervektors $\vec{v}$ und den Vektor $\vec{q}$ folgt:

$$\begin{aligned} \|\vec{v} - \vec{q}\| &= \left\| \begin{pmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \\ \sum_{i=1}^{n} x_i a_i \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ T \end{pmatrix} \right\| \\ &= \sqrt{\underbrace{(2x_1 - 1)^2}_{1} + \underbrace{(2x_2 - 1)^2}_{1} + \ldots + \underbrace{(2x_n - 1)^2}_{1} + \underbrace{\left(\sum_{i=1}^{n} x_i a_i - T\right)^2}_{0}},\ \text{mit } x_i \in \{0, 1\} \\ &= \sqrt{\underbrace{1 + 1 + \cdots + 1}_{n\text{-mal}}} = \sqrt{n}. \end{aligned}$$

Dies zeigt, dass eine Lösung für das SUBSET SUM-Problem eine Lösung im oben konstruierten Gitter für CVP impliziert. Die Richtung $w \notin \text{SUBSET SUM} \rightarrow f(w) \notin \text{CVP}$ wird mithilfe der Kontraposition $f(w) \in \text{CVP} \rightarrow w \in \text{SUBSET SUM}$ gezeigt.

Sei $\vec{v} = (v_1, \ldots, v_n, v_{n+1})^T$ ein nichtverschwindender Gittervektor, sodass $\|\vec{v} - \vec{q}\| \leq \sqrt{n}$. Die Komponenten $v_1, v_2, \ldots, v_n$ des Gittervektors sind aufgrund der oben konstruierten Gitterbasis B gerade und somit gilt: $|v_i - 1| \geq 1$ mit $i \in \{1, 2, \ldots, n\}$. Die Distanz des Gittervektors $\vec{v}$ und dem Vektor $\vec{q}$ ergibt:

$$\|\vec{v} - \vec{q}\| = \sqrt{(v_1 - 1)^2 + (v_2 - 1)^2 + \cdots + (v_n - 1)^2 + (v_{n+1} - T)^2} \leq \sqrt{n}\,.$$

Die Ungleichung $\|\vec{v} - \vec{q}\| \leq \sqrt{n}$ kann somit nur gelten, wenn $v_{n+1} = T$ gilt, sodass $(v_{n+1} - T)^2 = 0$ und für alle $v_i \in \{0, 2\}$ mit $i \in \{1, 2, \ldots, n\}$ die Gleichung $(v_i - 1)^2 = 1$ erfüllt ist. Es wird kurz wiederholt, dass $v_i \neq 1$ für $i \in \{1, 2, \ldots, n\}$ gelten muss, da die Komponenten $v_1, v_2, \ldots, v_n$ zwingend gerade sind. Somit muss $\sum_{i=1}^{n}(v_i/2)a_i = \sum_{i=1}^{n} x_i a_i = T$ mit $x_i \in \{0, 1\}$ gelten.

□

Dies zeigt, dass CVP ein NP-hartes Problem ist. Eine ausführliche Analyse der „Schwierigkeit“ des „Closest Vector Problems“ und des „Approximate Closest Vector Problems“ findet sich bspw. in [MG02, S. 50 ff.].

Die Abbildung 4.2 zeigt weitere komplexitätstheoretische Ergebnisse für die Approximation der Gitterprobleme CVP und SVP. Dabei sind in Abhängigkeit des Approximationsfunktion $\gamma(n)$, wobei n dem Rang des Gitters entspricht, die entsprechenden Komplexitätsklassen der Approximationsprobleme $\text{CVP}_{\gamma(n)}$ bzw. $\text{SVP}_{\gamma(n)}$ zugeordnet.

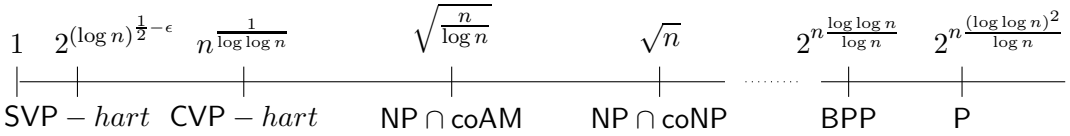


Abbildung 4.2: Komplexitätstherotische Analyse von Gitterproblemen (aus [AR05, S. 751]). Einige Konstanten wurden nicht angegeben.

4.3 Komplexität von SVP

Der Beweis der NP-Härte von CVP kann nicht direkt auf das SVP erweitert werden. Bisher konnte lediglich bewiesen werden, dass das SVP ein NP-hartes Problem unter sogenannten randomisierten (Karp-)Reduktionen ist (siehe [MG02, S. 56]). Ein Beweis für diese Aussage findet sich in [Ajt98].

Das SVP besitzt jedoch eine Eigenschaft, die für Kryptografieverfahren, die auf diesem Problem basieren, interessant ist. Damit ein Kryptografieverfahren als sicher gilt, muss dieses „average-case hard“ sein. Dies bedeutet, dass das Kryptografieverfahren sicher ist, sofern unter einer gegebenen Verteilung zufällig ein Schlüsselpaar bestehend aus öffentlichen und privaten Schlüssel ausgewählt werden. Im Gegensatz bedeutet „worst-case hard“, dass zur Verwendung des Kryptografieverfahren bestimmte „schwierige“ Schlüsselpaare ausgewählt werden müssen, sodass das unterliegende Berechnungsproblem nicht in Polynomialzeit gelöst werden kann, bspw. um aus dem öffentlichen Schlüssel den geheimen Schlüssel zu berechnen [MG02, S. 143]. Die NP-Härte eines Berechnungsproblems sagt bspw., dass es schwierig ist, alle Instanzen bzw. insbesondere die schwierigsten Instanzen („worst-case instances“) des Berechnungsproblems zu lösen. Es können also auch für NP-harte Probleme Instanzen existieren, die effizient, also in Polynomialzeit lösbar sind. Das Erfüllbarkeitsproblem ist bspw. NP-vollständig, aber es existieren bestimmte aussagenlogische Formeln, für die das Erfüllbarkeitsproblem in P liegt.

Die Konstruktion von asymmetrischen Kryptografieverfahren basiert auf „schwierigen“ Berechnungsproblemen. Derzeit existiert jedoch keine Möglichkeit, um beweisen zu können, dass ein Problem „schwierig“ ist, bspw. durch einen mathematischen Beweis [Ajt96, S. 1]. Es existiert kein Verfahren, um zu zeigen, dass eine Instanz eines Problems ein sogenannter „worst-case“ (auf Deutsch: schlechtester Fall) ist. Dies bedeutet, dass zur Lösung dieser Instanz die gemessene Ressource (in vielen Fällen die Laufzeit zur Berechnung der Lösung) ihr Maximum erreicht. Alle anderen Instanzen des gleichen Problems können also mit den Ressourcen, die von der „worst-case“ Instanz benötigt werden, gelöst werden. Eine zufällig ausgewählte Instanz des Berechnungsproblems unter Angabe der Verteilung der möglichen Probleminstanzen ist ein sogenannter „average-case“ (auf Deutsch: durchschnittlicher Fall) eines Problems. Es könnte aus diesem Grund sein, dass ein „average-case“ eines Problems einfacher zu lösen ist, als ein entsprechender „worst-case“ des gleichen Problems. Sollte ein Berechnungsproblem lediglich im „worst-case“ NP-hart, aber im „average-case“ wesentlich einfacher sein, so würde ein, auf diesem Berechnungsproblem basierendes Kryptografieverfahren in der Praxis als unsicher gelten.

Da kein Beweisverfahren für die Schwierigkeit des „average-case“ von Berechnungsproblemen existiert, bspw., dass eine zufällig ausgewählte („average-case“) Instanz des Problems NP-hart ist, wurden in der Vergangenheit andere Techniken benutzt, um „sichere“ Kryptografieverfahren zu konstruieren: Zum einen wurde versucht, NP-vollständige Probleme als Basis eines Kryptografieverfahrens zu benutzen. Ein Beispiel für einen solchen Versuch ist das durch Adi Schamir gebrochene Merkle-Hellman Knapsack Kryptografieverfahren [MH78][Sha84]. Es wird mittlerweile als unwahrscheinlich angesehen, dass sichere asymmetrische Kryptografieverfahren auf NP-vollständigen Problemen basieren können [Lee90, S. 749][Pap97, S. 2][BB09, S. 18]. Eine weitere Möglichkeit, um ein sicheres Kryptografieverfahren zu konstruieren, ist, ein für mehrere Jahrzehnte untersuchtes und als „schwierig“

angesehenes Problem, für das kein effizienter Algorithmus gefunden werden konnte, als Basis eines Kryptografieverfahrens zu benutzen. Ein Beispiel für ein solches Kryptografieverfahren ist RSA, bei dem dieses Berechnungsproblem das Faktorisieren von großen zusammengesetzten Zahlen ist. Es ist unbekannt, ob für das Faktorisierungsproblem ein Polynomialzeitalgorithmus für klassische Computer existiert, oder welche Eigenschaften eine zusammengesetzte Zahl besitzen muss, damit diese besonders schwierig faktorisiert werden kann, also ein „worst-case“ des Faktorisierungsproblems darstellt.

Miklos Ajtai hat 1996 in [Ajt96] gezeigt, dass eine bestimmte Klasse von Gittern existiert, in der eine Lösung einer beliebigen Instanz des SVP in einem Gitter mit Dimension m eine Lösung aller Instanzen des SVP in diesem Gitter mit Dimension n bis auf einen polynomiellen Ungenauigkeitsfaktor impliziert, wobei $m \gg n$ gilt. Dies wird als sogenannte „worst-case to average-case reduction“ bezeichnet und erleichtert die Analyse von gitterbasierten Problemen in einer solchen Klasse von Gittern. Anstatt alle Instanzen des SVP in einem Gitter mit Dimension n zu untersuchen, muss lediglich eine beliebige Instanz in einem Gitter mit der Dimension m der gleichen Klasse analysiert werden. Diese Idee und die benötigten Grundlagen werden im Folgenden vorgestellt.

Definition 4.8 *Ein* Orakel *f ist eine totale „Black Box“-Funktion, die für jedes Wort $w \in \Sigma^*$, den Funktionswert bzw. die Lösung $f(w)$ in einem Schritt berechnen kann. Ist die Funktion f für jede Eingabe $w \in \Sigma^*$ berechenbar, so kann ein solches Orakel auch als eine Art Unterprogramm angesehen werden [Yu07, S. 9 f.].*

Mithilfe von Orakeln kann das Konzept der Turingmaschine um ein Orakel erweitert werden. Dies führt u. a. zu zahlreichen neuen Komplexitätsklassen.

Definition 4.9 *Eine* Orakel-Turingmaschine *ist eine Turingmaschine, die während der Berechnung beliebig oft Anfragen an ein Orakel stellen kann. In der Notation wird eine Orakel-Turingmaschine wie folgt ausgedrückt: Die Orakel-Turingmaschine M^A besitzt ein Orakel für das Problem A, dass für jedes $w \in A$ den Funktionswert $f(w)$ in einem Schritt berechnen kann [Yu07, S. 9 f.].*

Der Laufzeit- und Speicherbedarf einer Orakel-Turingmaschine ist nur durch den Laufzeit- bzw. Speicherbedarf der Turingmaschine bestimmt. Der Laufzeit- bzw. Speicherbedarf, der für die Abarbeitung der Anfrage vom Orakel benötigt wird, wird hierbei vernachlässigt [BDG95, S. 31][Mel00, S. 43].

Eine genaue Erweiterung von Turingmaschinen zu Orakel-Turingmaschinen (inklusive der benötigten Zustände und Bänder) findet sich bspw. in [Yu07, S. 9 f.], [BDG95, S. 31] und [Mel00, S. 43].

Definition 4.10 *Eine probabilistische Orakel-Turingmaschine mit polynomieller Laufzeit ist eine Orakel-Turingmaschine mit der Eigenschaft, dass jeder Nachfolgezustand zusätzlich von einem „Münzwurf“ abhängig ist. Für jede Konfiguration der Turingmaschine existieren somit genau zwei Übergänge, aus denen zufällig und gleichwahrscheinlich anhand eines „Münzwurfs“ einer der beiden möglichen Übergänge ausgewählt wird. Eine probabilistische Orakel-Turingmaschine akzeptiert, wenn die Berechnung der Turingmaschine in einem akzeptierenden Zustand endet [BDG95, S. 137]. Aufgrund der „Münzwürfe“ ist jedoch nicht garantiert, dass bei einem erneuten Durchlauf und gleicher Eingabe die Turingmaschine das gleiche Ergebnis liefert.*

Definition 4.11 *Ein Problem A ist* turing-reduzierbar *auf ein Problem B, falls das Problem A mithilfe einer probabilistischen Orakel-Turingmaschine M^B in polynomieller Laufzeit entschieden werden kann, die ein Orakel für die Sprache B besitzt und diesem Orakel beliebig oft Anfragen stellen kann. Dieser Zusammenhang zwischen den Problemen A und B wird mithilfe der folgenden Schreibweise ausgedrückt:*

$$A \leq_{\mathrm{t}}^{\mathrm{p}} B\,.$$

[CJRR07, S. 571]

Definition 4.12 *Sei P ein beliebiges Problem, $\widehat{P}$ die Menge aller „worst-case“ Instanzen dieses Problems und P^* eine beliebige mit einer Verteilung $\mathcal{D}$ ausgewählte („average-case“) Instanz dieses Problems.*

Es gilt:

$\widehat{P} \leq_{\mathrm{t}}^{\mathrm{p}} P^*$ genau dann, wenn eine probabilistische Orakel-Turingmaschine M^A existiert, sodass für jedes Orakel A, das P^* löst, die Orakel-Turingmaschine M^A entsprechend alle Instanzen aus $\widehat{P}$ in polynomieller Laufzeit lösen kann.

Aus einer solchen Reduktion kann abgeleitet werden, dass wenn P^* einfach zu lösen ist, auch alle Instanzen $\widehat{P}$ einfach zu lösen sind. Die Umkehrung dieser Aussage bedeutet: Existiert mindestens eine Instanz in $\widehat{P}$, die schwierig zu lösen ist, so ist auch P^* schwierig zu lösen [CJRR07, S. 571]. Eine solche Reduktion wird als „worst-case to average-case reduction“ *bezeichnet.*

Miklos Ajtai hat in [Ajt96] gezeigt, dass eine solche „worst-case to average-case reduction“ für das SVP in einer bestimmten Klasse von Gittern existiert. Bevor die Beweisidee von Ajtai vorgestellt werden kann, muss zunächst ein weiteres Berechnungsproblem in Gittern vorgestellt werden.

Das „Shortest Vector Problem (SVP)“ kann wie folgt erweitert werden.

Definition 4.13 *Das „Shortest Indipendent Vector Problem (SIVP)“ ist wie folgt definiert:*

PROBLEM: *SIVP*
EINGABE: *Eine Gitterbasis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$.*
AUSGABE: *Eine Menge von n linear unabhängigen Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathcal{L}(B)$, sodass gilt:*

$$\max_{i=1}^{n} \|\vec{v}_i\| = \lambda_n \,.$$

Eine Variante dieses Problems ist wiederum die Approximation einer solchen Menge von n linear unabhängigen Vektoren in einem Gitter:

Definition 4.14 *Das „Approximate Shortest Indipendent Vector Problem ($SIVP_{\gamma(n)}$)“ ist wie folgt definiert:*

PROBLEM: $SIVP_{\gamma(n)}$
EINGABE: *Eine Basis $B \in \mathbb{Q}^{m \times n}$ mit $m \geq n$ eines Gitters $\mathcal{L}$.*
AUSGABE: *Eine Menge von n linear unabhängigen Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathcal{L}(B)$, sodass gilt:*

$$\lambda_n \leq \max_{i=1}^{n} \|\vec{v}_i\| \leq \gamma(n) \cdot \lambda_n \,.$$

Bei diesem Approximationsproblem $\mathsf{SIVP}_{\gamma(n)}$ ist $\gamma(n) \geq 1$ wiederum eine Funktion für die Güte der berechneten Lösung im Vergleich zur optimalen Lösung in Abhängigkeit von n, wobei n in vielen Fällen dem Rang des Gitters entspricht.

Anmerkung: Im Folgenden bezeichnet $1\mathsf{SVP}^m$ eine beliebige Instanz des „Shortest Vector Problem“ in einem Gitter mit Dimension m, das unter einer Verteilung $\mathcal{D}$ ausgewählt wurde und ${}^*\mathsf{SIVP}^n_{p(n)}$ alle Instanzen des „Approximate Shortest Indipendent Vector Problem“ in einem Gitter mit Dimension n, wobei für die Approximationsfunktion $\gamma(n) =_{\text{def}} p(n) =_{\text{def}} \mathcal{O}(n^c)$ gilt, wobei c eine beliebige Konstante ist.

Miklos Ajtai hat den folgenden Zusammenhang zwischen ${}^*\mathsf{SIVP}^n_{p(n)}$ und $1\mathsf{SVP}^m$ gezeigt:

$${}^*\mathsf{SIVP}^n_{p(n)} \leq^{\text{p}}_{\text{t}} 1\mathsf{SVP}^m, \text{ wobei } m \gg n \,,$$

indem er eine probabilistische Orakel-Turingmaschine $M^{1\mathsf{SVP}^m}$ angibt, sodass diese mit einem Orakel für $1\mathsf{SVP}^m$ entsprechend ${}^*\mathsf{SIVP}^n_{p(n)}$ in polynomieller Laufzeit lösen kann.

Im Folgenden werden die Ideen des Beweises basierend auf [Ajt96] und [GGH96] vorgestellt: Für den Beweis wird eine bestimmte Klasse von Gittern benötigt, die wie folgt definiert ist (nach [Ajt96, S. 4 f.])

Definition 4.15 *Sei $M \in \mathbb{Z}_q^{n \times m}$ mit $m \geq n$ eine Matrix, so bilden alle Vektoren $\vec{v} \in \mathbb{Z}_q^m$ mit $M\vec{v} \equiv \vec{0} \pmod{q}$ ein Gitter $\mathcal{L}(B)$ mit Rang n und Dimension m. Die m-dimensionalen Gittervektoren $\vec{v} \in \mathcal{L}(B)$ besitzen dabei als Komponenten Elemente aus der Menge $\{0, 1, \ldots, q-1\}$. Es ist nicht möglich eine einfache Form einer Gitterbasis für den allgemeinen Fall eines solchen Gitters anzugeben. Es ist jedoch möglich, die Basis zu einem solchen Gitter in polynomieller Laufzeit zu berechnen [MG02, S. 18].*

Ist die Addition $\vec{v} = \vec{u} + \vec{w}$ zweier Vektoren $\vec{u}, \vec{w} \in \operatorname{span}(B)$ ein Gittervektor mit $M\vec{v} \equiv M(\vec{u}+\vec{w}) \equiv \vec{0} \pmod{q}$, so muss entweder $\vec{u} \in \mathcal{L}(B)$ und $\vec{w} \in \mathcal{L}(B)$ oder $\vec{u} \notin \mathcal{L}(B)$ und $\vec{w} \notin \mathcal{L}(B)$ gelten.

Es wird kurz erwähnt, dass der m-dimensionale Nullvektor $\vec{0}$ ein Gittervektor ist, da $M\vec{0} \equiv \vec{0} \pmod{q}$ gilt.

Satz 4.16 *Sei $m, n, q \in \mathbb{N}$ mit $n \log(q) < m < \frac{q}{2n^4}$ und $q \in \mathcal{O}(n^c)$, wobei $c > 0$ eine Konstante ist, dann erzeugt die Menge*

$$\{\vec{v} : M\vec{v} \equiv 0 \pmod{q}\}, \text{ wobei } M \in \mathbb{Z}_q^{n \times m} \text{ mit } m \geq n$$

eine bestimmte Klasse von Gittern $\mathcal{L}(B)$, in denen

$$^*\mathsf{SIVP}^n_{p(n)} \leq^{\mathrm{p}}_{\mathrm{t}} 1\mathsf{SVP}^m,$$

mit $\gamma(n) \in \mathcal{O}(n^d)$, wobei d eine Konstante ist, gilt.

Aus dieser Reduktion lässt sich auch ein kürzester Gittervektor für $^*\mathsf{SVP}^n_{p(n)}$ ableiten, weshalb nur die obere Reduktion gezeigt wird [Ajt96, S. 2 und 6]. Für weitere Eigenschaften und die Struktur von solchen speziellen Gittern siehe [MG02, S. 147 f.].

Anmerkung: Eine Grundmasche kann, wie in Satz 2.21 gezeigt, die lineare Hülle $\operatorname{span}(B)$ des Gitters $\mathcal{L}(B)$ in unendliche viele Grundmaschen aufteilen. Im Folgenden wird die Grundmasche einer kürzesten und „nahezu orthogonalen“ Gitterbasis als *Basisgrundmasche* bezeichnet. Die Abbildungen 4.3 und 4.4 verdeutlichen anhand eines zweidimensionalen Beispiels ($m = n = 2$) eine solche Unterteilung der linearen Hülle des Gitters mithilfe von unterschiedlichen Grundmaschen bzw. zugehörigen Gitterbasen.

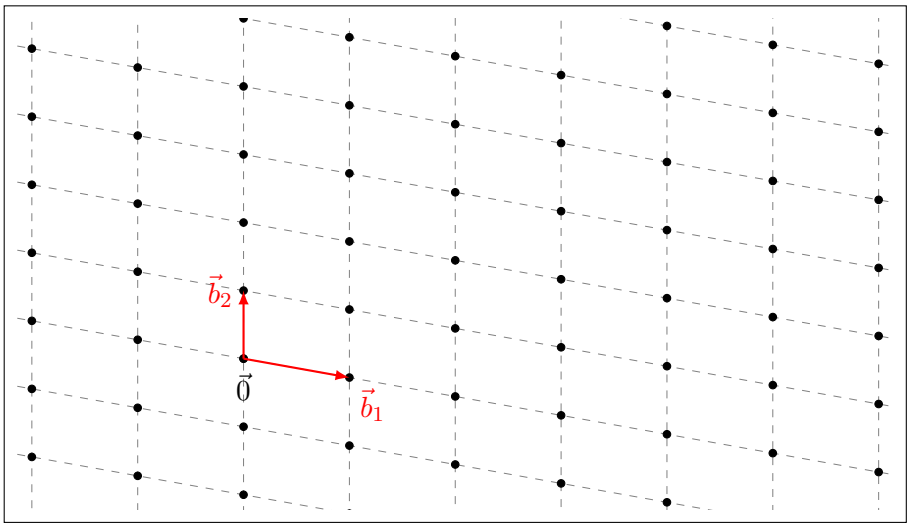


Abbildung 4.3: Eine Unterteilung des Vektorraums $\mathbb{R}^2$ anhand der Basisgrundmasche.

Beweisidee:

Die beiden Berechnungsprobleme SVP und $\mathsf{SIVP}_{\gamma(n)}$ werden noch einmal für diese, in diesem Beweis benutzten, bestimmten Gitter ohne Verwendung der Gitterbasis wiederholt:

Definition 4.17

PROBLEM: *SVP'*

EINGABE: *Eine Matrix* $M \in \mathbb{Z}_q^{n \times m}$ *mit* $n \log(q) < m < \frac{q}{2n^4}$ *und* $q \in \mathcal{O}(n^c)$, *wobei* $c > 0$ *eine Konstante ist.*

AUSGABE: *Ein kürzester nichtverschwindender Vektor* $\vec{v} \in \mathbb{Z}_q^m$, *sodass* $M\vec{v} \equiv \vec{0} \pmod{q}$ *mit* $\|\vec{v}\| \leq n$.

Definition 4.18

PROBLEM: $\mathit{SIVP'}_{\gamma(n)}$

EINGABE: *Eine Matrix* $M \in \mathbb{Z}_q^{n \times m}$ *mit* $n \log(q) < m < \frac{q}{2n^4}$ *und* $q \in \mathcal{O}(n^c)$, *wobei* $c > 0$ *eine Konstante ist.*

AUSGABE: *Eine Menge von* n *linear unabhängigen Gittervektoren* $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathbb{Z}_q^m$, *sodass*

$$M\vec{v}_1 \equiv M\vec{v}_2 \equiv \cdots \equiv M\vec{v}_n \equiv 0 \pmod{q}$$

und

$$\lambda_n \leq \max_{i=1}^{n} \|\vec{v}_i\| \leq \gamma(n) \cdot \lambda_n \,,$$

mit $\gamma(n) \in \mathcal{O}(n^d)$, *wobei* d *eine Konstante ist, gilt.*

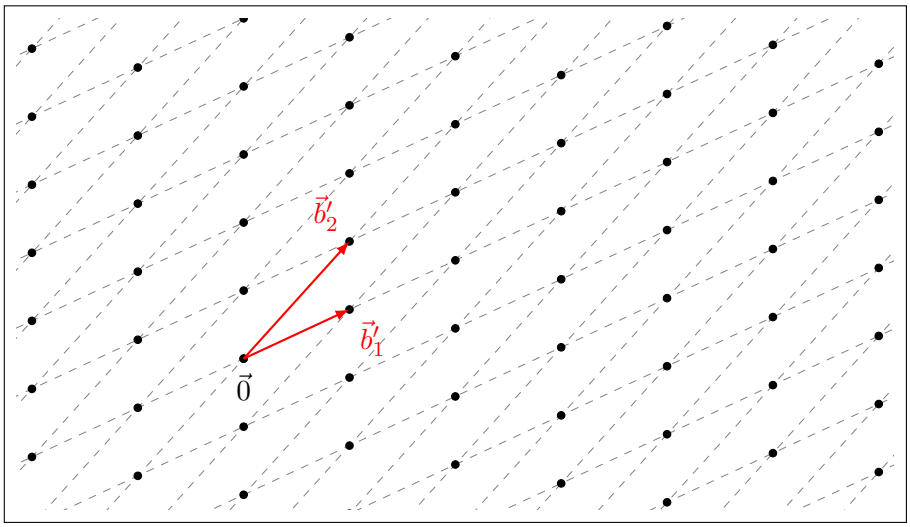


Abbildung 4.4: Eine Unterteilung des Vektorraums $\mathbb{R}^2$ anhand einer beliebigen Grundmasche.

Die hier vorgestellte Reduktion zeigt, dass alle Instanzen des „Approximative Shortest Indipendent Vector Problem“ mit einen maximalen polynomiellen Ungenauigkeitsfaktor in einem Gitter der Dimension n mithilfe eines Orakels für eine beliebige Instanz des „Shortest Vector Problem“ in einem Gitter mit Dimension m und $m \gg n$ in polynomieller Laufzeit gelöst werden kann. Die Reduktion benutzt als Eingabe die n Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ des Gitters und wählt dann zufällig n linear unabhängige Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$, mit der Eigenschaft, dass $\|\vec{v}_1\| \leq \|\vec{v}_2\| \leq \cdots \leq \|\vec{v}_n\|$ gilt. Aus dieser Menge von n linear unabhängigen Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$ kann dann (mithilfe des Orakels) ein Gittervektor $\vec{w}$ erzeugt werden, der linear unabhängig zu $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{n-1}$ ist und $\|\vec{w}\| \leq \frac{\|\vec{v}_n\|}{2}$ gilt. Da der Gittervektor $\vec{w}$ linear unabhängig zu den Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{n-1}$ ist, kann der Gittervektor $\vec{v}_n$ durch den Gittervektor $\vec{w}$ ersetzt werden. Dieser Vorgang kann rekursiv solange wiederholt werden, bis die Ungleichung

$$\max_{i=1}^{n} \|\vec{v}_i^*\| \leq \max_{i=1}^{n} \|\vec{v}_i\| \leq \gamma(n) \cdot \max_{i=1}^{n} \|\vec{v}_i^*\|,$$

mit $\gamma(n) \in \mathcal{O}(n^d)$, wobei d eine Konstante ist, erfüllt ist. Dabei sind die Gittervektoren $\vec{v}_1^*, \vec{v}_2^*, \ldots, \vec{v}_n^*$ die exakte Lösung des „Shortest Indipendent Vector Problems“ des entsprechenden Gitters. Es wird angenommen, dass diese Ungleichung beim Beginn der Reduktion noch nicht erfüllt ist. Der Beweis besteht aus insgesamt fünf Schritten. Dabei dienen die ersten vier Schritte zur Konstruktion einer beliebigen Instanz („average-case“) des „Shortest Vector Problem“ in einem Gitter der Dimension m. Der letzte Schritt zeigt dann, wie mithilfe einer Lösung für das „Shortest Vector Problem“ in diesem Gitter der Dimension m alle Instanzen (insbesondere auch die „worst-case“ Instanzen) des „Approximate Shortest Indipendent Vector Problems“ in einem Gitter mit Dimension n in polynomieller Laufzeit bis auf einen polynomiellen Fehler gelöst werden können.

Es wird wiederum versucht, die einzelnen Schritte der Beweisidee anhand von zweidimensionalen Beispielen mit $n = 2$ und $q = 3$ zu verdeutlichen. Die Abbildung 4.5 zeigt das Beispielgitter, welches zur Verdeutlichung des Beweises benutzt wird.

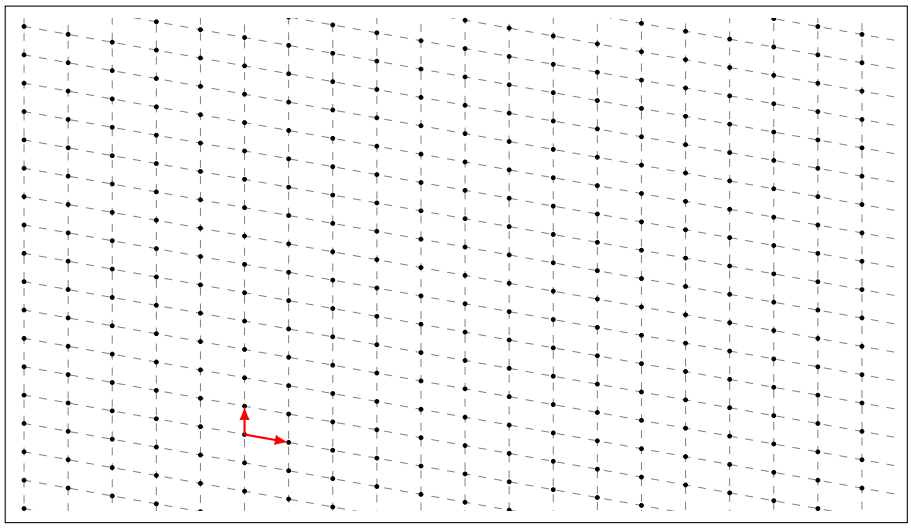


Abbildung 4.5: Das zweidimensionale Beispielgitter ($n = 2, q = 3$), das in einem m-dimensionalen Raum eingebettet ist und zur Veranschaulichung des Beweises benutzt wird.

1. **Erstellung eines „Pseudowürfels“:** Der erste Schritt besteht daraus, aus n linear unabhängigen Gittervektoren ein n-dimensionales Spat zu konstruieren, das „nahezu“ einem n-dimensionalen Hyperwürfel entspricht. Dieses Spat wird im Folgenden deshalb als „Pseudowürfel“ bezeichnet.

   Die Konstruktion des Pseudowürfels funktioniert wie folgt: Zuerst wird ein n-dimensionaler Hyperwürfel C mit einer beliebigen Seitenlänge a im Vektorraum $\mathbb{R}^n$ aufgespannt. Die Vektoren $a \cdot \vec{e}_1, a \cdot \vec{e}_2, \ldots, a \cdot \vec{e}_n \in \mathbb{R}^n$ seien die Vektoren, die diesen n-dimensionalen Hyperwürfel aufspannen, wobei $\vec{e}_i$ der i. Einheitsvektor im Vektorraum $\mathbb{R}^n$ ist. Im Folgenden wird angenommen, dass die Seitenlänge des Würfels C ungefähr $n^3 \cdot \|\vec{v}_n\|$ entspricht. Die Abbildung 4.6 verdeutlicht den zweidimensionalen Würfel C im Vektorraum $\mathbb{R}^2$ für das zweidimensionale Beispielgitter.

   Die Vektoren $a \cdot \vec{e}_1, a \cdot \vec{e}_2, \ldots, a \cdot \vec{e}_n \in \mathbb{R}^n$ sind nicht notwendigerweise Gittervektoren. Der nächste Schritt ist, diese Vektoren bzw. die Ecken des n-dimensionalen Hyperwürfels auf Gittervektoren abzubilden. Hierzu werden die Vektoren $a \cdot \vec{e}_i$, für $i \in \{1, 2, \ldots, n\}$ als reelle Linearkombination der vorher zufällig ausgewählten Vektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$ beschrieben. Mithilfe von Babais ROUNDING OFF-PROCEDURE werden die Vektoren $a \cdot \vec{e}_i$ für $i \in \{1, 2, \ldots, n\}$ auf die Gittervektoren $\vec{f}_i$ abgebildet, sodass gilt:

$$a \cdot \vec{e}_i = r_{1_i}\vec{v}_1 + r_{2_i}\vec{v}_2 + \cdots + r_{n_i}\vec{v}_n, \text{ mit } r_{1_i}, r_{2_i}, \ldots, r_{n_i} \in \mathbb{R}\,.$$

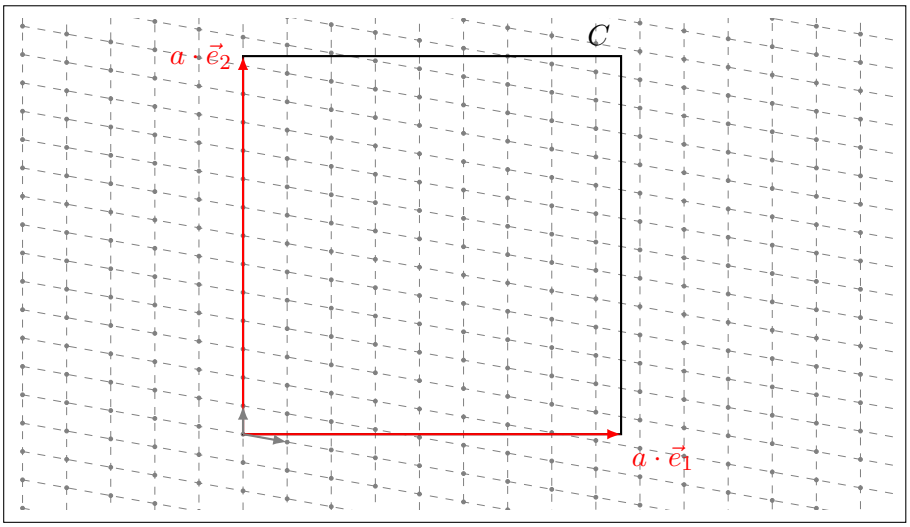


Abbildung 4.6: Überlagerung eines Würfels C in dem Gitter $\mathcal{L}$.

$$\vec{f}_i = \lfloor r_{1_i} \rceil \vec{v}_1 + \lfloor r_{2_i} \rceil \vec{v}_2 + \cdots + \lfloor r_{n_i} \rceil \vec{v}_n, \text{ mit } \lfloor r_{1_i} \rceil, \lfloor r_{2_i} \rceil, \ldots, \lfloor r_{n_i} \rceil \in \mathbb{Z}\,.$$

Die resultierenden Gittervektoren $\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n$ spannen den Pseudowürfel $\widetilde{C}$ auf. Die Differenz der Ecken des Würfels C zu den Ecken des Pseudowürfels $\widetilde{C}$ beträgt maximal $n \cdot \|\vec{v}_n\|$. Die Abbildung 4.7 zeigt den so konstruierten Pseudowürfel für das zweidimensionale Beispiel.

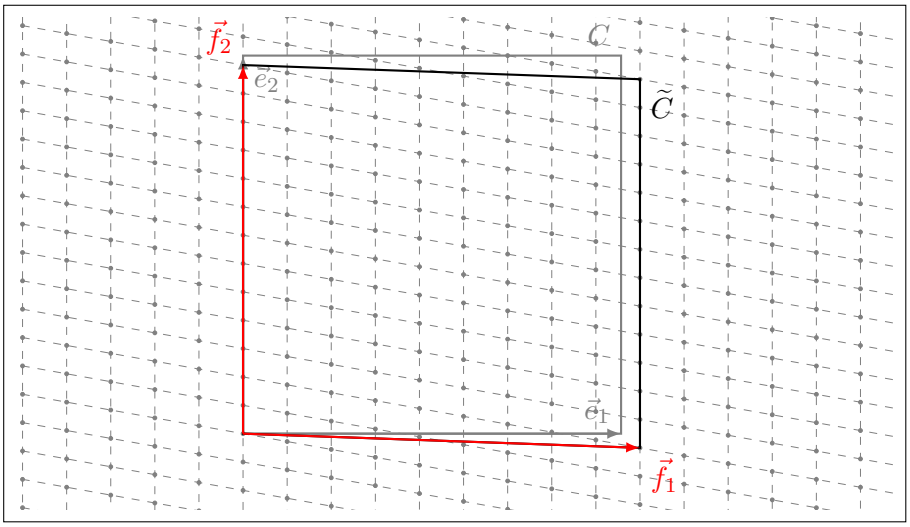


Abbildung 4.7: Annäherung eines Pseudowürfels $\widetilde{C}$ an den Würfel C mit Babais Heuristik.

2. **Unterteilung des Pseudowürfels in Teil-Pseudowürfel:** Im zweiten Schritt wird der Pseudowürfel $\widetilde{C}$ in q^n gleichgroße Teil-Pseudowürfel $\widetilde{C}_{\vec{t}}$ unterteilt, die wie

folgt durch einen Vektor $\vec{T} \in \mathbb{Z}_q^n$ dargestellt werden können:

$$C_{\vec{T}} =_{\text{def}} \left\{ \sum_{i=1}^{n} r_i \vec{f}_i : \frac{t_i}{q} \leq r_i < \frac{t_i + 1}{q} \right\} \text{ mit } \vec{T} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix} \in \mathbb{Z}_q^n .$$

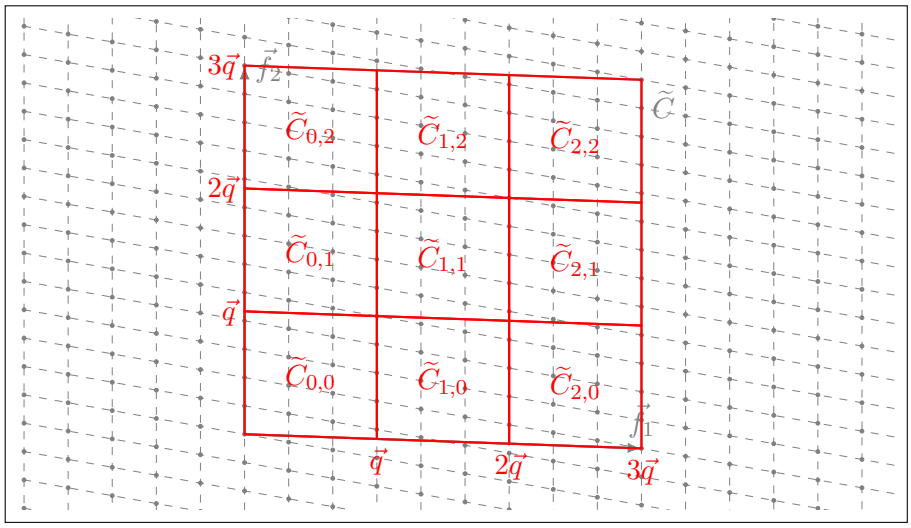


Abbildung 4.8: Aufteilung des Pseudowürfels in q^n gleich große Teil-Pseudowürfel ($n = 2$ und $q = 3$).

Für jeden Teil-Pseudowürfel $\widetilde{C}_{\vec{T}}$ wird der Vektor $\vec{\theta}_{\vec{T}} \in \widetilde{C}_{\vec{T}}$ wie folgt definiert, sodass $\|\vec{\theta}_{\vec{T}} - \vec{0}\| < \|\vec{\alpha} - \vec{0}\|$ für alle $\vec{\alpha} \in (\widetilde{C}_{\vec{T}} \backslash \{\vec{\theta}_{\vec{T}}\})$ gilt:

$$\vec{\theta}_{\vec{T}} =_{\text{def}} \sum_{i=1}^{n} \frac{t_i}{q} \vec{f}_i .$$

Die Vektoren $\vec{\theta}_{\vec{T}}$ werden auch als *Ursprung* des Teil-Pseudowürfels $\widetilde{C}_{\vec{T}}$ bezeichnet, da dieser Vektor innerhalb des Teil-Pseudowürfels $\widetilde{C}_{\vec{T}}$ am nächsten zum Ursprung bzw. dem Nullvektor $\vec{0}$ ist. Die Abbildung 4.9 zeigt die Ursprungsvektoren der Teil-Pseudowürfel anhand des zweidimensionalen Beispiels.

Jeder Vektor $\vec{s} \in \widetilde{C}_{\vec{T}}$ kann eindeutig in zwei Vektoren $\vec{\theta}_{\vec{T}}$ und $\vec{\alpha}$ zerlegt werden, sodass $\vec{s} = \vec{\theta}_{\vec{T}} + \vec{\alpha}$ gilt, wobei $\vec{\alpha} \in (\widetilde{C}_{\vec{T}} \backslash \{\vec{\theta}_{\vec{T}}\})$. Die Seitenlänge jedes Teil-Pseudowürlfels $\widetilde{C}_{\vec{T}}$ ist ungefähr $\frac{n^3 \cdot \|\vec{v}_n\|}{q}$. Daraus folgt, dass die Seitenlänge jedes Teil-Pseudowürfels $\widetilde{C}_{\vec{T}}$ weiterhin „viel länger“ als der längste Basisvektor ist, der die Basisgrundmasche erzeugt. Aus dieser Eigenschaft folgt, dass jeder Teil-Pseudowürfel $\widetilde{C}_{\vec{T}}$ ungefähr gleich viele Gitterpunkte beinhaltet [Ajt96, S. 7][GGH96, S. 37].

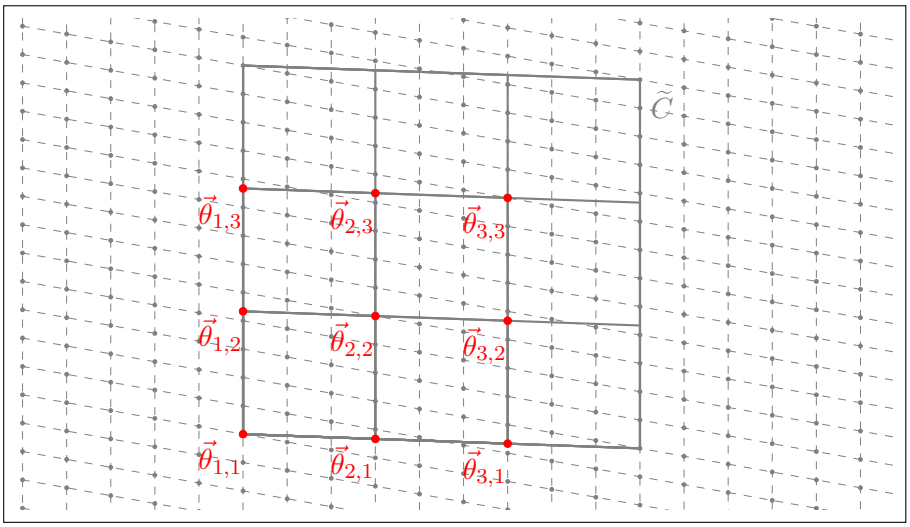


Abbildung 4.9: Die Ursprungsvektoren $\vec{\theta}_{\vec{T}}$ der Teil-Pseudowürfel. Jeder Teil-Pseudowürfel enthält genau 13 Gittervektoren.

3. **Wählen von beliebigen Gittervektoren im Pseudowürfel $\widetilde{C}$:** Im dritten Schritt werden zufällig m unterschiedliche Gittervektoren $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m$ im Pseudowürfel $\widetilde{C}$ gewählt.

   Diese m unterschiedlichen Gittervektoren werden wie folgt zufällig gewählt, sodass garantiert ist, dass diese im Pseudowürfel $\widetilde{C}$ enthalten sind. Seien die Gittervektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ die Basisvektoren des Gitters, so werden zufällig m Gittervektoren $\vec{p}_i, \vec{p}_2, \ldots, \vec{p}_m$ als Linearkombination der Basisvektoren $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n$ mit ganzzahligen Koeffizienten im Intervall $[0, 2^{c \cdot n} \cdot \max(S, \|\vec{b}_1\|, \|\vec{b}_2\|, \ldots, \|\vec{b}_n\|)]$ gewählt, wobei c eine Konstante ist. Für jeden Gittervektor $\vec{p}_i$ mit $i \in \{1, 2, \ldots, m\}$ gilt:

$$\vec{p}_i = x_{1_i}\vec{b}_1 + x_{2_i}\vec{b}_2 + \cdots + x_{n_i}\vec{b}_n \,,$$

   mit $x_{1_i}, x_{2_i}, \ldots, x_{n_i} \in [0, 2^{c \cdot n} \cdot \max(S, \|\vec{b}_1\|, \|\vec{b}_2\|, \ldots, \|\vec{b}_n\|)]$. Diese Gittervektoren $\vec{p}_i, \vec{p}_2, \ldots, \vec{p}_m$ müssen noch so transformiert werden, dass diese auf Gittervektoren innerhalb des Pseudowürfels $\widetilde{C}$ abgebildet werden. Aus diesem Grund wird jeder dieser m Gittervektoren $\vec{p}_i$ mit $i \in \{1, 2, \ldots, m\}$ durch $\vec{p}_i \pmod{\widetilde{C}}$ auf einen Gittervektor innerhalb des Pseudowürfels $\widetilde{C}$ abgebildet.

   Die Reduktion „$\vec{p}_i \pmod{\widetilde{C}}$“ funktioniert wie folgt: Die Punkte $\vec{p}_i, \vec{p}_2, \ldots, \vec{p}_m$ werden als reelle Linearkombination der Vektoren $\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n$ dargestellt. Für jeden Gittervektor $\vec{p}_i$ mit $i \in \{1, 2, \ldots, m\}$ gilt dann:

$$\vec{p}_i = r_{1_i}\vec{f}_1 + r_{2_i}\vec{f}_2 + \cdots + r_{n_i}\vec{f}_n, \text{ mit } r_{1_i}, r_{2_i}, \ldots, r_{n_i} \in \mathbb{R} \,.$$

Der Gittervektor $\vec{p}_i$ wird dann wie folgt auf einen Gittervektor $\vec{u}_i \in \widetilde{C}$ abgebildet:

$$\vec{u}_i = (r_{1_i} - \lfloor r_{1_i} \rfloor)\vec{f}_1 + (r_{2_i} - \lfloor r_{2_i} \rfloor)\vec{f}_2 + \cdots + (r_{n_i} - \lfloor r_{n_i} \rfloor)\vec{f}_n \,,$$

mit $(r_{1_i} - \lfloor r_{1_i} \rfloor), (r_{2_i} - \lfloor r_{2_i} \rfloor), \ldots, (r_{n_i} - \lfloor r_{n_i} \rfloor) \in [0, 1)$.

Die reellen Koeffizienten der Linearkombination vom Gittervektor $\vec{p}_i$ werden somit auf den Anteil nach dem Komma reduziert. Da die reellen Koeffizienten $(r_{1_i} - \lfloor r_{1_i} \rfloor), (r_{2_i} - \lfloor r_{2_i} \rfloor), \ldots, (r_{n_i} - \lfloor r_{n_i} \rfloor)$ der Linearkombination vom Vektor $\vec{u}_i$ im Intervall $[0, 1)$ liegen, muss der Vektor $\vec{u}_i$ im Pseudowürfel $\widetilde{C}$ enthalten sein.

Der Vektor $\vec{u}_i$ ist ein Gittervektor, da

$$\vec{u}_i = (r_{1_i} - \lfloor r_{1_i} \rfloor)\vec{f}_1 + (r_{2_i} - \lfloor r_{2_i} \rfloor)\vec{f}_2 + \cdots + (r_{n_i} - \lfloor r_{n_i} \rfloor)\vec{f}_n$$

$$= \underbrace{r_{1_i}\vec{f}_1 + r_{2_i}\vec{f}_2 + \cdots + r_{n_i}\vec{f}_n}_{\text{Gittervektor } \vec{p}_i} - \underbrace{\lfloor r_{1_i} \rfloor \vec{f}_1 - \lfloor r_{2_i} \rfloor \vec{f}_2 - \cdots - \lfloor r_{n_i} \rfloor \vec{f}_n}_{\text{Gittervektor, da} \lfloor r_{1_i} \rfloor, \lfloor r_{2_i} \rfloor, \ldots, \lfloor r_{n_i} \rfloor \in \mathbb{Z}} \,,$$

und die Subtraktion von Gittervektoren abgeschlossen ist, also wiederum einen Gittervektor ergibt. Die Abbildung 4.10 verdeutlicht die Reduktion eines beliebigen Gittervektors in den Pseudowürfel $\widetilde{C}$ anhand des zweidimensionalen Beispiels.

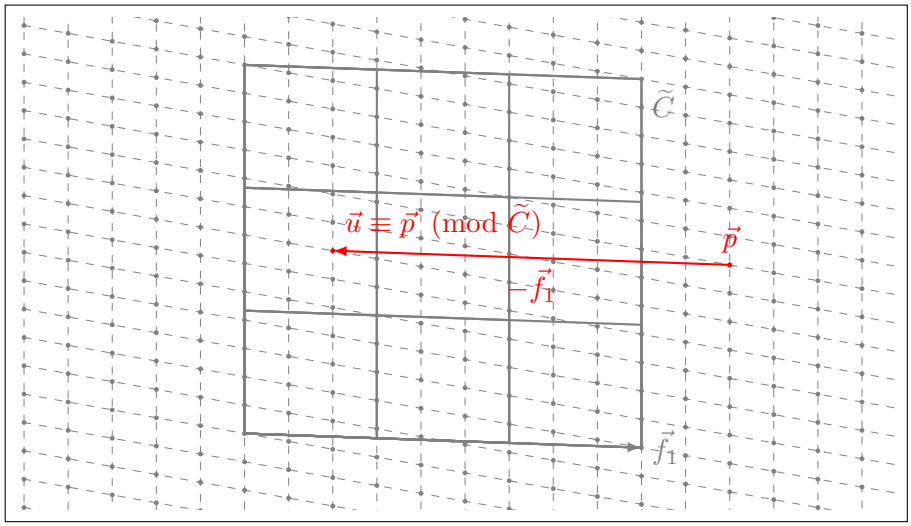


Abbildung 4.10: Beispiel für eine Reduktion eines beliebigen Gittervektors $\vec{p} \pmod{\widetilde{C}}$.

Sind die ganzzahligen Koeffizienten für die Linearkombinationen der Gittervektoren $\vec{p}_i, \vec{p}_2, \ldots, \vec{p}_m$ groß genug, so entspricht die Auswahl dieser Gittervektoren laut [GGH96, S. 38] „nahezu“ der Gleichverteilung.

4. **Erzeugung einer Instanz vom „average-case“ des „Shortest Vector Problems“:** Im vierten Schritt wird aus den m Gittervektoren $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m$ im Pseudowürfel $\widetilde{C}$ eine Instanz des „average-case“ des „Shortest Vector Problems“ erzeugt.

Für jeden der Gittervektoren $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m$ wird der Vektor $\vec{T}_1, \vec{T}_2, \ldots, \vec{T}_m \in \mathbb{Z}_q^n$ berechnet, sodass für jeden Gittervektor $\vec{u}_i \in \widetilde{C}_{\vec{T}_i}$ mit $i \in \{1, 2, \ldots, m\}$ gilt. Es wird also bestimmt, in welchem der q^n Teil-Pseudowürfel der Gittervektor $\vec{u}_i$ enthalten ist.

Da die Gittervektoren $\vec{u}_i$ „nahezu“ gleichverteilt sind und alle Teil-Pseudowürfel „nahezu“ die gleiche Anzahl an Gittervektoren besitzen, folgt laut [GGH96, S. 38] daraus, dass auch die Vektoren $\vec{T}_i \in \mathbb{Z}_q^n$ „nahezu“ gleichverteilt sind.

Aus den Vektoren $\vec{T}_1, \vec{T}_2, \ldots, \vec{T}_m \in \mathbb{Z}_q^n$ wird eine Matrix M konstruiert, die diese Vektoren als Spaltenvektoren besitzt:

$$M =_{\text{def}} \left[\vec{T}_1 | \vec{T}_2 | \cdots | \vec{T}_m\right] .$$

Die Matrix $M \in \mathbb{Z}_q^{n \times m}$ mit $m \geq n$ ist, da die Vektoren $\vec{T}_1, \vec{T}_2, \ldots, \vec{T}_m$ „nahezu“ gleichverteilt ausgewählt wurden, eine „nahezu“ gleichverteilte Instanz von $1\mathsf{SVP}^m$.

5. **Berechnung eines kürzesten Gittervektors:** Im fünften Schritt wird mithilfe des konstruierten „average-case“ des „Shortest Vector Problems“ in einem Gitter der Dimension m das „Shortest Indipendent Vector Problem“ in einem Gitter der Dimension n bis auf einen polynomiellen Ungenauigkeitsfaktor gelöst. Angenommen die Lösung der im vierten Schritt erzeugten Instanz von $1\mathsf{SVP}^m$ ergibt durch Anwendung eines Orakels einen Vektor $\vec{x} \in \mathbb{Z}_q^m$, mit $M\vec{x} \equiv \sum_{i=1}^m x_i \vec{T}_i \equiv 0 \pmod q$ und die Wahrscheinlichkeit, dass für die Länge des Vektors $\|\vec{x}\| \leq n$ gilt, sei mindestens $\frac{1}{2}$. Aus diesem kürzesten Vektor $\vec{x} \in \mathbb{Z}_q^m$ kann ein Gittervektor $\vec{w}$ als Linearkombination der Gittervektoren $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_m$ wie folgt berechnet werden:

$$\vec{w} = x_1\vec{u}_1 + x_2\vec{u}_2 + \ldots x_m\vec{u}_m = \sum_{i=1}^{m} x_i\vec{u}_i ,$$

wobei $x_1, x_2, \ldots, x_m \in \mathbb{Z}_q^m$ die Komponenten des Vektors $\vec{x}$ sind. Im zweiten Schritt wurde gezeigt, dass jeder Vektor $\vec{s}$ in einem der Teil-Pseudowürfel $\widetilde{C}_T$ eindeutig in zwei Vektoren $\vec{\theta}_{\vec{T}}$ und $\vec{\alpha}$ zerlegt werden kann. Somit können die Gittervektoren $\vec{u}_i$ für $i \in \{1, 2, \ldots, n\}$ wie folgt zerlegt werden:

$$\vec{u}_i = \vec{\theta}_{\vec{T}_i} + \vec{\alpha}_i \text{ mit } \vec{\theta}_{\vec{T}_i} =_{\text{def}} \sum_{i}^{n} \frac{t_i}{q}\vec{f}_i \text{ und } \vec{\alpha} \in (\widetilde{C}_{\vec{T}_i} \backslash \{\vec{\theta}_{\vec{T}_i}\}).$$

Für den Gittervektor $\vec{w}$ folgt daraus:

$$\vec{w} = \sum_{i=1}^{m} x_i\vec{u}_i = \sum_{i=1}^{m} x_i\vec{\theta}_{\vec{T}_i} + \sum_{i=1}^{m} x_i\vec{\alpha}_i .$$

Wenn dieser Gittervektor $\vec{w}$ auf einen Gittervektor innerhalb des Pseudowürfels ab-

gebildet wird, also $\vec{w} \pmod{\widetilde{C}}$ berechnet wird, so gilt $\sum_{i=1}^{m} x_i \vec{\theta}_{\vec{T}_i} \equiv \vec{0} \pmod{\widetilde{C}}$. Im Folgenden bezeichnet $\vec{T}_i(j)$ die j. Komponente des Vektors $\vec{T}_i$. Aus $\sum_{i=1}^{m} x_i \vec{\theta}_{\vec{T}_i}$ folgt:

$$\sum_{i=1}^{m} x_i \vec{\theta}_{\vec{T}_i} = \sum_{i=1}^{m} x_i \sum_{j=1}^{n} \frac{\vec{T}_i(j)}{q} \vec{f}_i = \sum_{j=1}^{n} \overbrace{\frac{\sum_{i=1}^{m} x_i \vec{T}_i(j)}{q}}^{\vec{x} \cdot „j. \text{ Zeile von } M“ \equiv 0 \pmod{q} \Rightarrow c_j \cdot q} \vec{f}_i = \sum_{j=1}^{n} \frac{c_j \cdot q}{q} \vec{f}_j = \sum_{j=1}^{n} c_j \vec{f}_j \,.$$

Dabei gilt $c_j \in \mathbb{Z}$ für $j \in \{1, 2, \ldots, n\}$. Für die Abbildung dieses Vektor in den Pseudowürfel $\widetilde{C}$ folgt daraus:

$$\left(\sum_{j=1}^{n} c_j \vec{f}_j \right) \bmod \widetilde{C} = \sum_{j=1}^{n} (c_j - \lfloor c_j \rfloor) \vec{f}_j = \vec{0}\,, \text{ da } c_j \in \mathbb{Z} \text{ und somit } \lfloor c_j \rfloor = c_j \text{ gilt.}$$

Für $\vec{w} \pmod{\widetilde{C}}$ folgt:

$$\vec{w} \equiv \sum_{i=1}^{m} x_i \vec{\theta}_{\vec{T}_i} + \sum_{i=1}^{m} x_i \vec{\alpha}_i \equiv \vec{0} + \sum_{i=1}^{m} x_i \vec{\alpha}_i \equiv \sum_{i=1}^{m} x_i \vec{\alpha}_i \pmod{\widetilde{C}}\,,$$

Da $\vec{\alpha}_i$ in einem Teil-Pseudowürfel enthalten ist und dieser nach Schritt 3 eine Seitenlänge von $\frac{n^3 \|\vec{v}_n\|}{q}$ besitzt, so kann der Vektor $\vec{\alpha}$ maximal eine Länge von $\frac{n \cdot n^3 \|\vec{v}_n\|}{q} = \frac{n^4 \|\vec{v}_n\|}{q}$ besitzen.

Da $\|\vec{x}\| = \sqrt{\langle x, x \rangle} \leq n$ gilt, folgt daraus $|\langle x, x \rangle| = |\sum_{i=1}^{m} x_i^2| \leq n^2$ für $i \in \{1, 2, \ldots, n\}$. Die Länge des Gittervektors $\vec{w} \pmod{\widetilde{C}}$ beträgt maximal:

$$\left\| \sum_{i=1}^{m} x_i \vec{\alpha}_i \right\| \leq \sum_{i=1}^{m} |x_i| \cdot \|\vec{\alpha}_i\| \leq n^2 \cdot \frac{n^4 \|\vec{v}_n\|}{q} = \frac{1}{q} \cdot n^6 \|\vec{v}_n\| \underbrace{\leq}_{q \geq n^7} \frac{\|\vec{v}_n\|}{2}.$$

Laut [Ajt96, S. 8] ist der Gittervektor $\vec{w}$ mit sehr hoher Wahrscheinlichkeit linear unabhängig zu den Gittervektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{n-1}$. Diese Prozedur kann mit der neuen Menge von n linear unabhängigen Gittervektoren $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{n-1}\} \cup \{\vec{w}\}$ wiederholt werden, bis n linear unabhängige Vektoren $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$ gefunden wurden, welche die Ungleichung

$$\max_{i=1}^{n} \|\vec{v}_i^*\| \leq \max_{i=1}^{n} \|\vec{v}_i\| \leq \gamma(n) \cdot \max_{i=1}^{n} \|\vec{v}_i^*\|\,,$$

mit $\gamma(n) \in \mathcal{O}(n^d)$, wobei d eine Konstante ist, erfüllen. Dabei sind die Gittervektoren $\vec{v}_1^*, \vec{v}_2^*, \ldots, \vec{v}_n^*$ die exakte Lösung des „Shortest Indipendent Vector Problems“ des entsprechenden Gitters.

Dies zeigt, dass in einer solchen Klasse von Gittern zum Lösen aller Instanzen des SIVP in einem Gitter mit Dimension n eine zufällige ausgewählte Instanz des SVP in einem Gitter der Dimension m und $m \gg n$ der gleichen Klasse gelöst werden muss. Eine effiziente

Lösung des SVP in einer solchen Klasse von Gittern mit Dimension m würde somit einen effizienten Algorithmus implizieren, der jede Instanz des SVP in einem Gitter in der gleichen Klasse mit Dimension n effizient lösen bzw. bis auf einen maximalen polynomiellen Faktor approximieren kann [BB09, S. 150].

Diese Eigenschaft der „worst-case to average-case reduction“ ist einzigartig für die gitterbasierten Kryptografieverfahren [MG02, S. x][BB09, S. 150] und zeigen, dass diese interessante Kandidaten für die Post-Quantum Kryptografie sind, da derzeit auch keine Quantenalgorithmen existieren, welche die unterliegenden Gitterprobleme erheblich effizienter lösen können, als entsprechende Algorithmen für klassische Computer [BB09, S. 151].

Da diese „worst-case to average-case reduction“ bedeutet, dass für die Untersuchung aller Instanzen des SVP in Gittern der Dimension n lediglich eine beliebige Instanz des SVP in einem Gitter mit Dimension m (innerhalb der gleichen Klasse) untersucht werden muss, wurde von Richard Lindner und Markus Rückert die sogenannte „Lattice Challenge“ veröffentlicht. Diese „Lattice Challenge“ und eine kurze Einführung findet sich unter `http://latticechallenge.org`.

Weitere Eigenschaften und eine ausführliche Analyse des „Shortest Vector Problems“ findet sich in [MG02, S. 69 ff.]. Eine chronologische Auflistung u. a. zu komplexitätstheoretischen Ergebnissen von Gitterproblemen und offene Probleme in diesem Gebiet gibt [Sch08, S. 73 ff.].

Kapitel 5

Gitterbasierte Kryptografieverfahren

In diesem Kapitel wird kurz ein kryptografisches Verfahren beschrieben, welches auf einem der vorgestellten Gitterproblemen basiert. Dabei wird für das kryptografische Verfahren lediglich die Idee hinter diesem Verfahren beschrieben. Des Weiteren wird erläutert, welches Berechnungsproblem ein potenzieller *Angreifer* lösen muss, um das entsprechende Verfahren zu brechen. Für etwaige mathematische Details (bspw. Beweise für die Wahl der einzelnen Parameter des kryptografischen Verfahren) wird weiterführende Literatur angegeben.

Hinweis: In diesem Kapitel wird der Begriff „beweisbar sicher“ benutzt. Der Begriff „beweisbar sicher“ bedeutet in diesem Kontext, dass für das Kryptpgrafieverfahren bzw. das unterliegende Berechnungsproblem eine „worst-case to average-case reduction“ gefunden werden kann.

Die kryptografischen Verfahren, die auf Gitterproblemen aufbauen, können grob in zwei Bereiche aufgeteilt werden: Es existieren Verfahren, deren unterliegendes Gitterproblem „beweisbar sicher“ ist. Diese Verfahren sind meist langsam, dafür aber existiert eine „worst-case to average-case reduction“ für das unterliegende Gitterproblem. Da diese Eigenschaft der „beweisbaren Sicherheit“ lediglich für das SVP gezeigt werden konnte und bisher keine direkte Reduktion von CVP zu SVP gefunden wurde, basieren solche kryptografische Verfahren auf dem SVP. Im Gegensatz zu diesen „beweisbar sicheren“ Verfahren existieren gitterbasierte Kryptografieverfahren, für deren Sicherheit kein solcher Beweis existiert [MG02, S. 184]. Diese Verfahren sind in der Regel effizienter und basieren meist auf dem CVP.

In der Einleitung wurde bereits erwähnt, dass zur Konstruktion eines asymmetrischen Kryptografieverfahrens eine Einwegfunktion bzw. genauer eine Einwegfunktion mit *Falltür* benötigt wird. Die „worst-case to average-case reduction“ von Ajtai kann als Klasse

von Einwegfunktionen aufgefasst werden [MG02, S. 161]. Diese Klasse von Einwegfunktionen besitzen jedoch keine *Falltürinformation*, sodass diese Einwegfunktionen nicht effizient invertiert werden können. In [GGH96] konnte jedoch aufbauend auf dieser Klasse von Einwegfunktionen eine „beweisbar sichere“ Klasse von kollisionsresistenten Hashfunktionen konstruiert werden. Das Finden von Kollisionen innerhalb einer Hashfunktion in dieser Klasse würde eine Lösung für die Approximation eines kürzesten Vektors (im verwendeten Gitter) bis auf einen polynomiellen Ungenauigkeitsfaktor implizieren.

Im Jahr 1997 wurden zwei unterschiedliche Verschlüsselungsverfahren vorgestellt, die auf Gitterproblemen basieren. Zum einen das Ajtai-Dwork Kryptografieverfahren, für das eine „worst-case to average-case reduction“ gezeigt werden konnte. Dieses Kryptografieverfahren verschlüsselt eine gegebenene Nachricht Bit für Bit [AD97, S. 284] und ordnet jedem Bit einen Vektor in der linearen Hülle eines n-dimensionalen Gitters zu [Dwo97, S. 5], sodass der Speicherbedarf der verschlüsselten Nachricht um einen polynomiellen Faktor in Abhängigkeit der zu verschlüsselnden Bitlänge ansteigt. Damit dieses Kryptografieverfahren als „praktisch sicher“ gilt, muss $n > 32$ gelten, was u. a. eine Größe des öffentlichen Schlüssels von ca. 20MB impliziert. Dieses Verfahren ist somit ohne weitere Effizienzsteigerungen nur aus theoretischen Gründen interessant [NS00, S. 10]. Das zweite im Jahr 1997 vorgestellte Kryptografieverfahren stammt von Oded Goldreich, Shafi Goldwasser und Shai Halevi und entspricht einer Einwegfunktion mit *Falltür* [GGH97, S. 113], die auf der „Schwierigkeit“ des CVP basiert. Die Einwegfunktion benutzt die Tatsache, dass dieses Problem mithilfe einer „guten“ Gitterbasis (bestehend aus kurzen und nahezu orthogonalen Basisvektoren) einfach gelöst werden kann. Diese „gute“ Gitterbasis ist die *Falltürinformation*, die für die Invertierung der Einwegfunktion benötigt wird. Für eine „schlechte“ Gitterbasis (bestehend entsprechend aus langen Basisvektoren) ist dieses Problem und auch die Approximation des Problems bis auf einen polynomiellen Ungenauigkeitsfaktor schwierig. Diese Einwegfunktion mit *Falltür* wurde ebenfalls in [GGH97] zu einem Verschlüsselungsverfahren erweitert, indem eine mögliche Abbildung einer beliebigen Nachricht auf einen Vektor in der linearen Hülle des Gitters angegeben wurde. Dieses Verschlüsselungsverfahren kann mit dem McEliece Kryptografieverfahren verglichen werden, nur dass in Gittern gerechnet wird [GGH97, S. 114][BB09, S. 167].

Im Folgenden wird kurz die Ver- bzw. Entschlüsselung mithilfe der in [GGH97] vorgestellten Einwegfunktion beschrieben. Dabei wird die Erzeugung von privaten und öffentlichen Schlüssel nur formal beschrieben. Eine detaillierte Beschreibung der Erzeugung des Schlüsselpaars findet sich in [GGH97, S. 119 ff.].

- Der private Schlüssel ist eine „gute“ Basis des Gitters. Der Begriff „gute Basis“ bezeichnet hier eine Gitterbasis mit möglichst kurzen und nahezu orthogonalen Basisvektoren. Für solche Basen kann, wie bereits in Kapitel 3 erläutert, das CVP effizient gelöst werden (bspw. mit Babais ROUNDING OFF-PROCEDURE), wenn der

Vektor $\vec{q} \in \text{span}(B)$ zu dem der nächste Gittervektor gesucht wird, nicht zu weit von einem Gitterpunkt entfernt ist.

- Der öffentliche Schlüssel ist eine „schlechte“ Gitterbasis. Diese besteht entsprechend aus Basisvektoren, die lang und paarweise einen möglichst kleinen Winkel zwischen sich besitzen. Micciancio hat in [Mic01] vorgeschlagen, dass für eine solche „schlechte“ Gitterbasis die Hermite Normal Form (siehe z. B. [Mic01, S. 3]) einer beliebigen Gitterbasis B benutzt wird. Diese eindeutige Basis eines Gitters kann aus jeder Gitterbasis effizient berechnet werden. Daraus folgt, dass keine Rückschlüsse auf die verwendete „gute“ Gitterbasis, die als privater Schlüssel benutzt wird, möglich sind.

- Zur Verschlüsselung werden zwei Vektoren benötigt: Ein Gittervektor $\vec{v} \in \mathcal{L}(B)$ und ein Fehlervektor $\vec{r} \in (\text{span}(B) \backslash \mathcal{L}(B))$, der in seiner euklidischen Länge begrenzt ist. Eine beliebige zu verschlüsselnde Nachricht wird auf den Fehlervektor $\vec{r}$ abgebildet und der Gittervektor $\vec{v}$ wird zufällig gewählt. Aus diesen beiden Vektoren lässt sich die verschlüsselte Nachricht $\vec{q} = \vec{v} + \vec{r}$ erzeugen. Der Vektor $\vec{q}$, der die verschlüsselte Nachricht repräsentiert und in der linearen Hülle des Gitters $\mathcal{L}(B)$ liegt, kann kein Gittervektor sein.

- Um die verschlüsselte Nachricht zu entschlüsseln, muss zu dem Vektor $\vec{q}$ der nächste Gittervektor $\vec{v}$ gefunden werden, sodass $\|\vec{v} - \vec{q}\|$ minimal ist. Dies ist nur mithilfe des privaten Schlüssels effizient möglich, einer Gitterbasis, die aus kurzen und nahezu orthogonalen Basisvektoren besteht. Der Gittervektor $\vec{v}$ lässt sich effizient bspw. mithilfe von Babais ROUNDING OFF-PROCEDURE berechnen. Da der Fehlervektor $\vec{r} \in (\text{span}(B) \backslash \mathcal{L}(B))$ die Nachricht beinhaltet, muss $\vec{r} = \vec{q} - \vec{v}$ berechnet werden und die Abbildung, die vor der Verschlüsselung die Nachricht auf den Fehlervektor $\vec{r}$ abgebildet hat, invertiert werden.

Damit ein *Angreifer* die verschlüsselte Nachricht $\vec{q} = \vec{v} + \vec{r}$ ohne Besitz des privaten Schlüssels entschlüsseln kann, muss dieser das CVP lösen. Eine Möglichkeit wäre die direkte Lösung des CVP unter Verwendung des öffentlichen Schlüssels. Eine weitere Angriffsmethode wäre aus dem öffentlichen Schlüssel den privaten Schlüssel zu berechnen (Gitterbasisreduktion) oder eine sehr gute Approximation des privaten Schlüssels. Als Gegenmaßnahme könnte hierfür auf den öffentlichen Schlüssel bereits ein Gitterbasisreduktionsalgorithmus angewendet werde (bspw. der LLL-Algorithmus), sodass eine weitere Anwendung dieses Algorithmus keine Verkürzung der Gitterbasis erreichen kann.

Sei n der Sicherheitsparamter des GGH-Kryptografieverfahrens, so muss der öffentliche Schlüssel die Länge $\mathcal{O}(n^2)$ betragen, damit das Verfahren als sicher gilt. Die Operationen, die zur Ver- bzw. Entschlüsselung berechnet werden müssen, besitzen einen Rechenaufwand von $\mathcal{O}(n^2)$ [GGH97, S. 112]. Als Gitterrang wird in [GGH97, S. 115] $n > 300$ vorgeschlagen. Es konnte jedoch in [Ngu99] gezeigt werden, dass für Gitter mit Rang $n = 350$

dieses Verfahren gebrochen werden kann. Die Idee des Angriffs in [Ngu99, S. 6 und 9 f.] ist, dass die Instanz des CVP, die bei der Verschlüsselung erzeugt wird, einfacher zu lösen ist, als eine zufällige Instanz des CVP. Dies resultiert aus der Tatsache, dass der Fehlervektor $\vec{r}$ viel kürzer als der zur Verschlüsselung benutzte Gittervektor $\vec{v}$ ist und der Fehlervektor des Weiteren eine bestimmte Form besitzen muss. In [Ngu99, S. 12 f.] und [Mic01] wird beschrieben, wie dieses Verfahren gehärtet werden kann, auch wenn gleichzeitig argumentiert wird, dass diese Maßnahmen das Verfahren ineffizient machen. Eine Zusammenfassung des Angriffs auf das GGH-Kryptosystem befindet sich in [Sch08, S. 49 ff.].

In [GGH97, S. 114] wird beschrieben, wie dieses Verschlüsselungsverfahren mithilfe des gleichen Schlüsselpaars als Signaturverfahren benutzt werden kann. Sei $\vec{q} \in \text{span}(B)$ ein Vektor, auf den die zu verschlüsselnde Nachricht vorher abgebildet wird. Die Signatur dieser Nachricht bzw. des Vektors, auf den diese Nachricht abgebildet wurde, besteht aus einem Paar von Vektoren $(\vec{q}, \vec{v})$. Neben dem öffentlichen Schlüssel (der „schlechten“ Gitterbasis) wird ein weiterer öffentlicher Parameter $\sigma \in \mathbb{R}$ benötigt. Die Signatur wird akzeptiert, falls $\vec{v} \in \mathcal{L}(B)$ ist und $\|\vec{v} - \vec{q}\| < \sigma$ gilt. Dies kann mithilfe der öffentlichen Gitterbasis effizient verifiziert werden. Da σ im Vergleich zur Länge eines kürzesten Vektors gewählt wird, ist $\vec{v}$ die Lösung des CVP für den Vektor $\vec{q}$. Für die Signierung einer beliebigen Nachricht muss somit das CVP gelöst werden, welches nur mithilfe einer „guten“ Gitterbasis möglich ist. Eine Angriffsmethode wäre u. a. wiederum eine Gitterbasisreduktion der öffentlichen Gitterbasis, um Nachrichten ohne Besitz der öffentlichen Gitterbasis signieren zu können. Eine Eigenschaft dieses Signaturverfahren ist, dass Signaturen von ähnlichen Nachrichten gleich sind, da diese Nachrichten bei einer deterministischen Abbildung von der zu signierenden Nachricht auf einen Vektor in der linearen Hülle des Gitters nicht weit voneinander entfernt sind. Da es nur selten Fälle gibt, in denen diese Eigenschaft erwünscht ist (siehe [GGH97, S. 114]), kann die zu signierende Nachricht bspw. vorher als Eingabe einer Hashfunktion dienen und anstatt der eigentlichen Nachricht wird stattdessen der aus der Nachricht erzeugte Hashwert signiert.

Auch dieses Verfahren kann erfolgreich *angegriffen* werden. In [NR06] konnte gezeigt werden, dass jede Signatur einen Teil des zum Signieren verwendeten privaten Schlüssels (die „gute“ Gitterbasis) verrät. Hierfür wird die folgende besondere Grundmasche definiert:

$$\mathcal{P}_{\frac{1}{2}}(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n) = \mathcal{P}_{\frac{1}{2}}(B) =_{\text{def}} \{s_1\vec{b}_1 + s_2\vec{b}_2 + \cdots + s_n\vec{b}_n\} \text{ mit } s_1, s_2, \ldots, s_n \in \left[-\frac{1}{2}, \frac{1}{2}\right],$$

wobei B dem privaten Schlüssel, der Gitterbasis bestehend aus kurzen und nahezu orthogonalen Basisvektoren, entspricht.

Der Mittelpunkt dieser speziellen Grundmasche $\mathcal{P}_{\frac{1}{2}}(B)$ liegt im Ursprung des Gitters und alle Gittervektoren, die zum Signieren von Nachrichten benutzt werden, sind in der Grundmasche $\mathcal{P}_{\frac{1}{2}}(B)$ enthalten [NR06, S. 5]. In [NR06, S. 14] wurde experimentell gezeigt, dass ungefähr n^2 Signaturen benötigt werden, um Rückschlüsse auf den verwendeten privaten

Schlüssel zu erhalten, wobei n dem Sicherheitsparameter des Kryptografieverfahrens und gleichzeitig dem Rang des Gitters entspricht.

Die Abbildung 5.1 zeigt ein zweidimensionales Beispiel für diese Grundmasche. Aus dieser Grundmasche kann der zum Signieren verwendete private Schlüssel approximiert werden.

Abbildung 5.1: Die Grundmasche $\mathcal{P}_{\frac{1}{2}}(B)$ eines zweidimensionalen Gitters.

Eine Gegenmaßnahme zur Härtung dieses Signaturverfahrens wäre, die einzelnen Einträge im privaten Schlüssel, der geheimen Gitterbasismatrix, zu vergrößern. Diese Maßnahme macht das Verfahren gleichzeitig ineffizient [NR06, S. 14].

Es existieren noch weitere Verschlüsselungsverfahren, die spezielle Gitter voraussetzen. Zur Vollständigkeit werden kurz das NTRU-Verfahren, welches sogenannte zyklische Gitter benutzt bzw. das LWE-Verfahren erwähnt, welches hingegen auf sogenannten idealen Gittern basiert.

Kapitel 6

Fazit

Dieses Kapitel fasst den Inhalt dieser Arbeit kurz zusammen und gibt einen kurzen Ausblick auf weiterführende Themen der gitterbasierten Kryptografie, die nicht innerhalb dieser Arbeit behandelt wurden.

Es wurden kurz die wichtigsten Grundlagen der asymmetrischen Kryptografie und der Post-Quantum Kryptografie vorgestellt. Des Weiteren wurde die Gittertheorie eingeführt, die zum grundlegenden Verständnis für die gitterbasierten Kryptografieverfahren und die Analyse dieser Verfahren, wie z. B. die Gitterbasisreduktion, benötigt wird. Es wurde analysiert, inwieweit zwei bekannte Gitterprobleme als „schwierig“ angesehen werden können und welche Approximationsalgorithmen existieren, um diese Probleme in polynomieller Zeit bis auf einen exponentiellen Ungenauigkeitsfaktor lösen zu können. Zum Schluss wurde ein asymmetrisches Kryptografieverfahren vorgestellt, welches sowohl zur Ver- bzw. Entschlüsselung von Informationen benutzt werden kann, als auch zur Signierung von Nachrichten. In beiden Fällen wurden für diese Verfahren auch die wichtigsten Ergebnisse der Kryptoanalyse vorgestellt.

Die gitterbasierte Kryptografie ist ein interessanter Kandidat für die sogenannte Post-Quantum Kryptografie. Die Eigenschaft der „beweisbaren Sicherheit“ des SVP vereinfacht die Analyse von kryptografischen Verfahren, die auf Instanzen des SVP mit dieser Eigenschaft basieren, erheblich.

Weiterführende Themen zur gitterbasierten Kryptografie, die innerhalb dieser Arbeit nicht behandelt wurden, sind u. a. Verfahren, die auf Gittern einer speziellen Form basieren, bspw. zyklische Gitter (NTRU) oder die sogenannten idealen Gitter (LWE). LWE („Learning with Errors“) ist eines der effizientesten gitterbasierten kryptografischen Verfahren, für das die „beweisbare Sicherheit“ gezeigt werden konnte [BB09, S. 172]. An diesem Verfahren bzw. dem entsprechenden LWE-Problem wird derzeit geforscht, um noch effizientere Varianten dieses Problems (bspw. Ring-LWE) finden zu können.

Die zahlreiche Literatur, die seit dem Artikel von Miklos Ajtai im Jahr 1996 veröffentlicht

wurde, deutet daraufhin, dass dieser Bereich der Kryptografie auch in Zukunft ausgiebig untersucht wird, u.a. um weitere effiziente Varianten von existierenden Gitterproblemen zu finden oder die existierenden Verfahren auf eine effiziente Implementierung auf spezieller Hardware zu untersuchen.

Anhang A

Implementierungen von Algorithmen

Die folgende Implementierung des LLL-Algorithmus wurde nicht bzgl. der Laufzeit optimiert. In [HPS08, S. 412] und [Bre11, S. 63] finden sich bspw. Erläuterungen zur Laufzeitoptimierung des LLL-Algorithmus bzw. ein optimierter Pseudocode des LLL-Algorithmus. Des Weiteren wurde der Algorithmus zur Schritt für Schritt-Analyse implementiert und nur für Gitter mit einer relativ kleinen Dimension getestet.

```
(* LLL-ALGORITHMUS *)
(* Autor: Patrick Vogt *)
(* Datum: 12.04.2013 *)
(*
  nach: Buch, Hoffstein, Pipher, Silverman - An Introduction to
  Mathematical Cryptography, Seite 411, Errata zum Buch, Hoffstein,
  Pipher, Silverman - An Introduction to Mathematical Cryptography,
  Seite 23 und Lenstra, Lenstra, Lovasz - Factoring Polynomials with
  Rational Coefficients, Seite 516 und 519
*)

(* EINGABEDATEN *)
(* base: Basis mit b_1,...,b_n als Zeilenvektoren *)
base =
      {
        {100,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},
        {101, 50, 10,  0,  0,  0,  0,  0,  0,  0,  0},
        { 50, 10, 20, 30,  0,  0,  0,  0,  0,  0,  0},
        { 10, 20, 10, 10,100,  0,  0,  0,  0,  0,  0},
```

```
20          {200, 10,  5,  1, 90, 99,  0,  0,  0,  0,  0},
21          {  1,  2,  3,  4,  5,   6  7,  0,  0,  0,  0},
22          {100, 99, 98, 97, 96, 95, 94, 93,  0,  0,  0},
23          {  1,  2,  3,  4,  5,  6,  7,  8,  9,  0,  0},
24          {  1,  2,  3,  4,  5,  6,  7,  8,  9, 10,  0},
25          {100,  1, 99,  2, 98,  3, 97,  4, 96,  5, 95},
26          { 99,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0}
27        };
28  (* n: Anzahl der Basisvektoren *)
29  n = Length[base];
30  (* Laufvariable i fuer den zu verarbeitenden Basisvektor *)
31  i = 2;
32  (*
33    gsbase: Liste zur Speicherung der Gram-Schmidt orthogonalisierten
34    Basis
35  *)
36  gsbase = List[];
37
38  (* VERARBEITUNG *)
39  (* der erste Basisvektor b_1 wird uebernommen *)
40  AppendTo[gsbase, base[[1]]];
41
42  (* Solange i kleiner gleich der Anzahl der Basisvektoren ist *)
43  While[i <= n,
44   (* Gram-Schmidt Berechnung *)
45
46   (*
47     Da eventuell die Reihenfolge der Basisvektoren innerhalb der Schleife
48     vertauscht wird, muss die Gram-Schmidt Orthogonalisierung (fuer eine
49     Teilmenge der Basisvektoren) wiederholt werden koennen und somit in
50     der Schleife berechnet werden
51   *)
52
53   (* b_i: i. Basisvektor *)
54   bi = base[[i]];
55   (* o_i: i. orthogonalisierter Basisvektor *)
56   oi = bi;
57
58   (*
59     innere Schleife fuer die Subtraktion der 1,...,j < i
```

```
60      orthogonalisierten Basisvektoren
61    *)
62    For[j = 1, j < i, j = j + 1,
63
64     (* o_j: j. Gram-Schmidt orthogonalisierter Basisvektor *)
65     oj = gsbase[[j]];
66
67     (* Berechnung des Gram-Schmidt Koeffizienten my_ij *)
68     myij = (bi . oj)/(oj . oj);
69
70     (* Teilerzeugung vom i. Gram-Schmidt Basisvektor *)
71     oi = oi - (myij*oj);
72
73     ]; (* End For *)
74
75    (*
76      i. Gram-Schmidt orthogonalisierten Basisvektor zu der Gram-Schmidt
77      Basis hinzufuegen (nur im ersten Durchlauf moeglich) oder austauschen
78    *)
79    gsbase = If[Length[gsbase] < i,
80               Append[gsbase, oi], ReplacePart[gsbase, i -> oi]];
81
82    (* Laengenreduktion der Basis base *)
83
84    (* b_i: i. Basisvektor *)
85    bi = base[[i]];
86
87    (* innere Schleife um den i. Basisvektor zu laengenreduzieren *)
88    For[j = i - 1, j >= 1, j = j - 1,
89
90     (* b_j: j. Basisvektor *)
91     bj = base[[j]];
92     (* o_j: j. Gram-Schmidt orthogonalisierter Basisvektor *)
93     oj = gsbase[[j]];
94
95     (* Berechnung des Gram-Schmidt Koeffizienten my_ij *)
96     myij = (bi . oj)/(oj . oj);
97
98     (* i. Basisvektor laengenreduzieren *)
99     bi = bi - (Round[myij]*bj);
```

```
100
101   ]; (* End For *)
102
103   (* i. laengenreduzierten Basisvektor in der Basis base austauschen *)
104   base = ReplacePart[base, i -> bi];
105
106   (*
107     bis hierhin ist fuer die Basisvektoren b_1,...,b_i die
108     Laengenreduziertheit sichergestellt. Jetzt muss noch die
109     Lovasz-Eigenschaft ueberprueft werden
110   *)
111
112   (* Ueberpruefung der Lovasz-Eigenschaft *)
113
114   (* b_i: i. Basisvektor *)
115   bi = base[[i]];
116   (* b_i-1: i-1. Basisvektor *)
117   bi1 = base[[i - 1]];
118   (* o_i: i. orthogonalisierter Basisvektor *)
119   oi = gsbase[[i]];
120   (* o_i-1: i-1. orthogonalisierter Basisvektor *)
121   oi1 = gsbase[[i - 1]];
122   (*
123     my_i,i-1: Gram-Schmidt Koeffizient vom i. Basisvektor und dem
124     i-1. Gram-Schmidt orthogonalisierten Basisvektor
125   *)
126   myii1 = (bi . oi1)/(oi1 . oi1);
127
128   (* Lovasz-Eigenschaft erfuellt? *)
129   If[Norm[oi + myii1*oi1]^2 >= (3/4)*Norm[oi1]^2,
130    (* Then *)
131    i = i + 1,
132    (* Else *)
133    (* vertausche Basisvektor b_i mit b_i-1 *)
134    base = ReplacePart[base, (i - 1) -> bi] ;
135    base = ReplacePart[base, i -> bi1] ;
136    (*
137      wenn wir den ersten Basisvektor vertauschen, muss auch der erste
138      Vektor in der Gram-Schmidt Basis ausgetauscht werden
139    *)
```

```
140   If[(i - 1) == 1, gsbase = ReplacePart[gsbase, 1 -> bi]];
141   (*
142     nun muessen im naechsten Schleifendurchlauf u.a. das
143     Gram-Schmidt Orthogonalisierungsverfahren, die Laengenreduktion und
144     die Ueberpruefung der Lovasz-Eigenschaft fuer i >= max(i-1,2)
145     wiederholt werden
146   *)
147   i = Max[i - 1, 2];
148
149   ]; (* End If *)
150
151  ]; (* End While *)
152
153 (* AUSGABE *)
154 (* LLL-reduzierte Basis base *)
155 base
```

Listing A.1: Der LLL-Algorithmus in Mathematica

Stichwortverzeichnis

Dieses Verzeichnis soll als Hilfestellung für den Leser zum Auffinden von einzelnen Begriffsdefinitionen dienen, die innerhalb dieser Arbeit eingeführt oder verwendet werden.

Die Stichwörter sind aufsteigend nach dem Alphabet angeordnet. Der Buchstabe ß wurde nach Vorbild des DUDEN wie ss und die deutschen Umlaute ä, ö, ü und äu entsprechend wie die nicht umgelauteten Vokale a, o, u und au bei der Einsortierung behandelt.

K

L

M

N

O

P

Q

R

S

T

U

V

Literaturverzeichnis

[AD97] Ajtai, Miklós ; Dwork, Cynthia: A public-key cryptosystem with worst-case/average-case equivalence. In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* ACM, 1997, S. 284–293 [zitiert auf S. 90]

[Ajt96] Ajtai, M.: Generating hard instances of lattice problems. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* ACM, 1996, S. 99–108 [zitiert auf S. 74, 75, 76, 78, 83 und 87]

[Ajt98] Ajtai, Miklós: The shortest vector problem in L 2 is NP-hard for randomized reductions. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing* ACM, 1998, S. 10–19 [zitiert auf S. 73]

[AR05] Aharonov, Dorit ; Regev, Oded: Lattice problems in NP ∩ coNP. In: *Journal of the ACM (JACM)* 52 (2005), Nr. 5, S. 749–765 [zitiert auf S. 73]

[Bab86] Babai, László: On Lovász'lattice reduction and the nearest lattice point problem. In: *Combinatorica* 6 (1986), Nr. 1, S. 1–13 [zitiert auf S. 48, 52 und 54]

[BB09] Bernstein, D.J. ; Buchmann, J.: *Post-Quantum Cryptography*. Springer London, Limited, 2009. – ISBN 9783540887027 [zitiert auf S. 3, 4, 6, 63, 74, 88, 90 und 95]

[BDG95] Balcázar, J.L. ; Díaz, J. ; Gabarró, J.: *Structural complexity*. Springer-Verlag, 1995 (EATCS monographs on theoretical computer science Bd. 1). – ISBN 978354058384X [zitiert auf S. 75 und 76]

[Bre11] Bremner, M.R.: *Lattice Basis Reduction: An Introduction to the LLL Algorithm and Its Applications*. CRC PressINC, 2011 (Chapman and Hall Pure and Applied Mathematics Series). – ISBN 9781439807026 [zitiert auf S. 97]

[BSMM12] Bronstein, I.N. ; Semendjajew, K.A. ; Musiol, G. ; Mühlig, H.: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2012. – ISBN 978–3–8171–2008–6 [zitiert auf S. 8]

[Buc09] Buchmann, J.: *Einführung in die Kryptographie.* Springer, 2009 (Springer-Lehrbuch). – ISBN 9783540744511 [zitiert auf S. 2]

[CJRR07] Charikar, Moses (Hrsg.) ; Jansen, Klaus (Hrsg.) ; Reingold, Omer (Hrsg.) ; Rolim, José D. P. (Hrsg.): *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings.* Bd. *4627.* Springer, 2007 (Lecture Notes in Computer Science). – ISBN 978–3–540–74207–4 [zitiert auf S. 76]

[Dwo97] Dwork, Cynthia: Positive applications of lattices to cryptography. In: *Mathematical Foundations of Computer Science 1997.* Springer, 1997, S. 44–51 [zitiert auf S. 90]

[Gal12] Galbraith, S.D.: *Mathematics of Public Key Cryptography.* Cambridge University Press, 2012. – ISBN 9781107013926 [zitiert auf S. 7, 51, 52, 53, 54 und 68]

[GGH96] Goldreich, Oded ; Goldwasser, Shafi ; Halevi, Shai: Collision-free hashing from lattice problems. In: *Electronic Colloquium on Computational Complexity (ECCC)* Bd. 3, 1996, S. 236–241 [zitiert auf S. 78, 83, 85, 86 und 90]

[GGH97] Goldreich, Oded ; Goldwasser, Shafi ; Halevi, Shai: Public-key cryptosystems from lattice reduction problems. In: *Advances in Cryptology—CRYPTO'97.* Springer, 1997, S. 112–131 [zitiert auf S. 90, 91 und 92]

[GJ79] GAREY, M.R.A. ; Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979 (A Series of Books in the Mathematical Sciences) [zitiert auf S. 71]

[Gol04] Goldreich, O.: *Foundations of Cryptography - Volume One - Basic Tools.* Cambridge University Press, 2004. – ISBN 0–511–04120–9 [zitiert auf S. 2]

[HPS08] Hoffstein, J. ; Pipher, J. ; Silverman, J.H.: *An Introduction to Mathematical Cryptography.* Springer, 2008 (Undergraduate Texts in Mathematics). – ISBN 9780387779942 [zitiert auf S. 18, 50, 68 und 97]

[Klo97] Klotzek, B.: *Analytische Geometrie und Lineare Algebra.* Deutsch, 1997. – ISBN 9783817115327 [zitiert auf S. 29]

[Lee90] Leeuwen, J.: *Handbook of Theoretical Computer Science: Algorithms and Complexity.* Elsevier Science Limited, 1990 (Handbook of Theoretical Computer Science Bd. 1). – ISBN 9780444880710 [zitiert auf S. 74]

[LLL82] Lenstra, Arjen K. ; Lenstra, Hendrik W. ; Lovász, László: Factoring polynomials with rational coefficients. In: *Mathematische Annalen* 261 (1982), Nr. 4, S. 515–534 [zitiert auf S. 67]

[Mel00] Melkebeek, D. van: *Randomness and Completeness in Computational Complexity*. Springer, 2000 (ACM Doctoral Dissertation Award Series) [zitiert auf S. 75]

[MG02] Micciancio, D. ; Goldwasser, S.: *Complexity of Lattice Problems: A Cryptographic Perspective*. Kluwer Academic, 2002 (The Kluwer International Series in Engineering and Computer Science). – ISBN 9780792376880 [zitiert auf S. 5, 7, 24, 27, 36, 38, 40, 43, 72, 73, 74, 78, 88, 89 und 90]

[MH78] Merkle, R. ; Hellman, M.E.: Hiding information and signatures in trapdoor knapsacks. In: *Information Theory, IEEE Transactions on* 24 (1978), Nr. 5, S. 525–530. – ISSN 0018–9448 [zitiert auf S. 74]

[Mic01] Micciancio, Daniele: Improving lattice based cryptosystems using the Hermite normal form. In: *Cryptography and Lattices*. Springer, 2001, S. 126–145 [zitiert auf S. 91 und 92]

[Ngu99] Nguyen, Phong: Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from crypto'97. In: *Advances in Cryptology—CRYPTO'99* Springer, 1999, S. 288–304 [zitiert auf S. 91 und 92]

[Ngu10] Nguyen, Phong Q.: *LLL Algorithm*. Springer Berlin Heidelberg, 2010. – ISBN 9783642022968 [zitiert auf S. 36]

[NR06] Nguyen, Phong Q. ; Regev, Oded: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In: *Advances in Cryptology-EUROCRYPT 2006*. Springer, 2006, S. 271–288 [zitiert auf S. 92 und 93]

[NS00] Nguyen, Phong Q. ; Stern, Jacques: Lattice reduction in cryptology: An update. In: *Algorithmic number theory*. Springer, 2000, S. 85–112 [zitiert auf S. 90]

[Pap97] Papadimitriou, Christos: NP-completeness: A retrospective. In: *Automata, languages and programming* (1997), S. 2–6 [zitiert auf S. 74]

[RSA78] Rivest, R.L. ; Shamir, A. ; Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. In: *Communications of the ACM* 21 (1978), Nr. 2, S. 120–126 [zitiert auf S. 2]

[RT33] Rademacher, H. ; Toeplitz, O.: *Von Zahlen und Figuren: Proben mathematischen Denkens für Liebhaber der Mathematik*. Zweite Auflage. Springer, 1933 [zitiert auf S. 72]

[Sch87] Schnorr, Claus-Peter: A hierarchy of polynomial time lattice basis reduction algorithms. In: *Theoretical computer science* 53 (1987), Nr. 2, S. 201–224 [zitiert auf S. 63]

[Sch08] Schönberger, Kay: *Gitter in der Kryptographie*, Humboldt-Universität zu Berlin, Diplomarbeit, 2008 [zitiert auf S. 36, 61, 62, 88 und 92]

[Sha84] Shamir, Adi: A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. In: *Information Theory, IEEE Transactions on* 30 (1984), Nr. 5, S. 699–704 [zitiert auf S. 74]

[Sho97] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: *SIAM Journal on Computing* 26 (1997), Nr. 5, S. 1484–1509 [zitiert auf S. VII und 3]

[Sip06] Sipser, M.: *Introduction To The Theory Of Computation*. Brooks Cole, 2006 (Computer Science Series). – ISBN 9780534950972 [zitiert auf S. 43, 69, 70 und 71]

[WP04] Wolf, C. ; Preneel, B.: Asymmetric cryptography: Hidden field equations. In: *European Congress on Computational Methods in Applied Sciences and Engineering*, 2004 [zitiert auf S. VII und 3]

[Yu07] Yu, F.: *On the Complexity of Real and Complex Functions Defined on the Two-dimensional Plane*. State University of New York at Stony Brook, 2007. – ISBN 9780549887041 [zitiert auf S. 75]

Inhalt des Datenträgers

- `./links` - beinhaltet eine Kopie der angegebenen Internetseiten.
- `./literature` - beinhaltet die verwendete Literatur untergliedert nach:
  - `./literature/books` - Bücher (falls vorhanden mit Springer-Link).
  - `./literature/lecture-notes` - Vorlesungsunterlagen.
  - `./literature/papers` - Wissenschaftliche Artikel.
  - `./literature/thesis` - Diplom- und Doktorarbeiten.
- `./pdf` - beinhaltet die Master-Arbeit im PDF-Format.
- `./src` - beinhaltet die in Mathematica implementierten Algorithmen.
- `./src-latex` - beinhaltet den LaTeX-Quellcode inkl. dem Quellcode für die Ti*k*Z-Abbildungen.
- `./Abstract.txt` - beinhaltet die Zusammenfassung dieser Arbeit im Textformat.