Konstruktion einer Gruppe mit unlösbarem Wortproblem

Jean-Baptiste Bellynck

München 2023

Konstruktion einer Gruppe mit unlösbarem Wortproblem

Jean-Baptiste Bellynck

Zulassungsarbeit
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Jean-Baptiste Bellynck
aus München

München, den 01.02.2023

Gutachter: Sebastian Wolfgang Hensel

Inhaltsverzeichnis

Abbildungsverzeichnis

Zusammenfassung

Das Wortproblem einer Präsention $G = \langle S \mid R \rangle$ ist die Frage, ob ein Wort aus dem Alphabet der Erzeuger S das neutrale Element ist. Es ist damit dem Wortproblem einer formalen Sprache sehr ähnlich. Dieses ist die Frage, ob ein Wort eines Alphabets Σ in einer formalen Sprache enthalten ist.

Das letztere Wortproblem ist in der Informatik gut erforscht. 1936 zeigte Alan Turing in [T$^+$36], dass es Sprachen gibt, bei denen das Wortproblem unlösbar ist. Für diese Sprachen ist es nicht möglich, festzustellen, ob ein bestimmtes Wort in der Sprache enthalten ist.

Diese Tatsache wirft die Frage auf, ob man diesen Befund aus der Informatik nutzen kann, um die Existenz einer endlich präsentierten Gruppe zu zeigen, die ein unlösbares Wortproblem hat. In der folgenden Arbeit sollen die Konfigurationen und Konfigurationsübergänge einer Turingmaschine mit unlösbarem Halteproblem (ein zum Wortproblem äquivalentes Problem) in einer Gruppe kodiert werden. Dabei wird die Eigenschaft des “Unlösbaren Wortproblems“ vererbt. Wir folgen dem Paper von Stephen G. Simpson [Sim05].
Durch Bass-Serre-Theorie nach [Ser02] schaffen wir eine Intuition für verwendete HNN-Erweiterungen und deren, durch Brittons Lemma gegebene, Normalform.

Danksagung. Ich möchte mich zuallererst bei meinem Betreuer Sebastian Hensel für seine verständnisvolle und zuvorkommende Unterstützung bedanken. Enthusiastisch führte er mich in das Gebiet der geometrischen Gruppentheorie ein und half mir, tiefer in die mathematische Welt einzusteigen.
Ich möchte mich zudem bei Ralf Gerkmann bedanken. Auch er motivierte mich, die Mathematik und insbesondere die Gruppentheorie stärker weiterzuverfolgen. Ohne ihn wäre diese Arbeit nicht entstanden.

Kapitel 1

Grundlagen der geometrischen Gruppentheorie

1.1 Freie Gruppen und Präsentationen

Wir beginnen damit, einige Grundlagen der geometrischen Gruppentheorie zu definieren. Als erstes definieren wir die freie Gruppe. Darauf aufbauend definieren wir Gruppenpräsentationen, eine mächtigere Version des Erzeugendensystems einer Gruppe. Präsentationen befreien uns von der Notwendigkeit, eine erzeugte Gruppe $\langle S \rangle \subseteq H$ als Untergruppe einer größeren Gruppe zu definieren, indem die Beziehungen zwischen den Erzeugern durch Relationen beschrieben werden.

Definition 1.1.1. Sei S eine Menge von Zeichen. Die *freie Gruppe über* S, notiert als $F(S)$ ist die Menge aller gekürzter Wörter, bestehend aus dem Alphabet S, und den dazu inversen Zeichen $S^{-1} := \{s_i^{-1} : s \in S\}$. D.h.

$$F_n := \left\{ w \in (S \sqcup S^{-1})^* : w \text{ ist gekürzt } \right\}$$

Ein Wort $w = w_1...w_n$ nennen wir *gekürzt*, wenn keine zwei aufeinanderfolgende Zeichen invers zueinander sind, d.h. $w_i \neq w_{i+1}^{-1}$.

Satz 1.1.2. *Die freie Gruppe ist eine Gruppe unter der Operation der Konkatenation und anschließenden Kürzung:*

$$w \circ v = w_1...w_n \circ v_1...v_m := w_1...w_{n-k}v_k...v_m$$
$$\text{für } w_{n-i} = v_i^{-1}, \forall i \in \{0, ..., k\}, w_{n-i-1} \neq v_{i+1}^{-1}$$

Das neutrale Element ist das leere Wort ε*, denn* $w \circ \varepsilon = w$.
Das zu einem Wort $w = w_1...w_n$ *inverse Element ist* $w^{-1} := w_n^{-1}...w_1^{-1}$*, denn* $w_1...w_n w_n^{-1}...w_1^{-1} = \varepsilon$

Satz 1.1.3. *Seien* S *und* T *zwei Alphabete mit gleich vielen Zeichen, d.h.* $|S| = |T|$*. Dann sind die freien Gruppen* $F(S)$ *und* $F(T)$ *bis auf Umbenennung der Erzeuger strukturgleich,*

d.h. $F(S) \cong F(T)$.
Um uns nicht immer auf ein Alphabet einigen zu müssen, bezeichnen wir mit F_n *die freie Gruppe über einem beliebigen Alphabet der Größe* n.

Beispiel 1.1.4. *Die freie Gruppe* F_1 *besteht aus allen gekürzten Wörtern des Alphabets* $\{s, s^{-1}\}$. *Ein Wort der Gruppe besteht entweder aus* $k \in \mathbb{N}$ *hintereinandergereihten* s *oder* s^{-1} *oder ist das leere Wort* ε.
Es gibt einen Gruppenisomorphismus $\varphi : F_1 \to \mathbb{Z}, \varphi(\varepsilon) = 0, \varphi(s^k) = k, \varphi((s^{-1})^k) = -k$. F_1 *ist somit isomorph zu* $\mathbb{Z}$.

Freie Gruppen sind nützlich, da sie Produktterme aus Erzeugern beschreiben. Das bedeutet: Schreibt man in einer beliebigen Gruppe ein Produkt aus Erzeugern $s_{i_1} \cdot ... \cdot s_{i_n}$, so gibt es für diesen Produktterm ein Element einer freien Gruppe, die den Term kodiert.

Definition 1.1.5. Gruppen können durch einen *Cayley-Graphen* visualisiert werden. Bei einem Cayley-Graphen steht jeder Knoten für ein Element der Gruppe. Jede gerichtete Kante steht für die Linksmultiplikation mit einem Erzeuger der Gruppe.

Cayley-Graphen sollen uns in der Arbeit lediglich eine Intuition zu speziellen Gruppen geben. Wir werden Cayley-Graphen daher nicht weiter formalisieren und bei Beispielen auch keine Herleitung angeben. Eine Einführung in Cayley-Graphen findet man in [Car21].

Der Cayley-Graph der freien Gruppe $F(\{r, b\})$ ist in Abbildung 1.1 zu sehen. Der mittlere Knoten repräsentiert die Identität. Durch Anmultiplizieren des Elements r folgt man in Pfeilrichtung einer roten Kante. Durch Anmultiplizieren des Elements b folgt man in Pfeilrichtung einer blauen Kante. Durch Anmultiplizieren des Elements r^{-1} folgt man gegen Pfeilrichtung der roten Kante. Durch Anmultiplizieren des Elements b^{-1} folgt man gegen Pfeilrichtung der blauen Kante.
Wir bezeichnen freie Gruppen als frei, da die Erzeuger im Gegensatz zu nicht-freien Gruppen in keiner algebraischen Beziehung zueinander stehen. Durch das Hinzufügen von algebraischen Beziehungen in Form von Gleichungen können wir neue Gruppen erhalten. Wie wir gleich sehen werden, lassen sich alle Gruppen auf diese Weise konstruieren.

Beispiel 1.1.6. *Wir fügen der freien Gruppe* $F_1 \cong F(s)$ *die Relation* $s^3 = \varepsilon$ *hinzu. Die Relation definiert eine Äquivalenzrelation auf den Wörtern aus* F_1. *Zwei Wörter* $v = v_1...v_n, w = w_1...w_m \in F_1$ *sind äquivalent, wenn man sie durch Anwendung von Relationen ineinander überführen kann.*
Unter der Anwendung von Relationen verstehen wir erstmal das Ersetzen von Zeichen gemäß der Relationsgleichungen. Durch das wiederholte Ersetzen von sss *durch* ε *zerfällt* F_1 *in die Äquivalenzklassen* $[\varepsilon] = \left\{s^{3k} : k \in \mathbb{Z}\right\}, [s] = \left\{s^{3k+1} : k \in \mathbb{Z}\right\}, [s^2] = \left\{s^{3k+2} : k \in \mathbb{Z}\right\}$.
Es gibt einen Isomorphismus von den Äquivalenzklassen in die zyklische Gruppe dritter Ordnung: $\varphi : F_1 \to C_3$ *mit* $\varphi([\varepsilon]) = 0, \varphi([s]) = 1, \varphi([s^2]) = 2$.

Beispiel 1.1.7. *Wir fügen der freien Gruppe* $F_2 \cong F(r, b)$ *die Relation* $rb = br$ *hinzu. Das bedeutet, dass der Produktterm* rb *und* br *als das gleiche Gruppenelement evaluieren. Im*

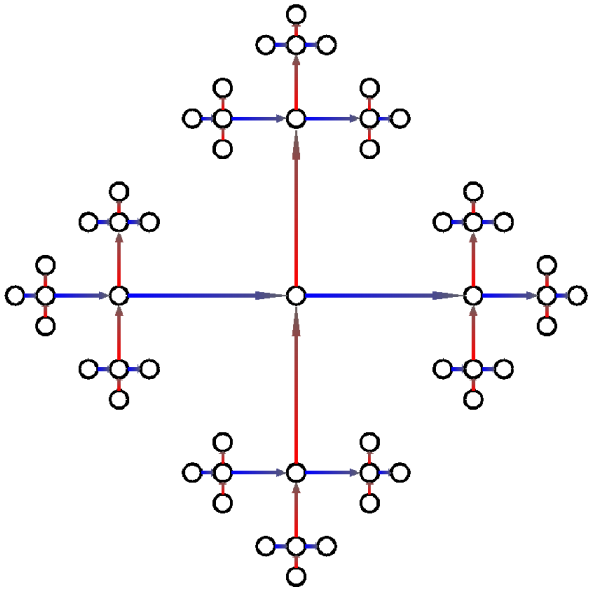


Abbildung 1.1: Endlicher Ausschnitt des Cayley-Graphen der Gruppe $F(\{r, b\}) \cong F_2$

Cayley-Graph (Abbildung 1.2) spiegelt sich das dadurch wieder, dass die beiden Pfade rb und br zum gleichen Knoten führen.
Dabei spielt es keine Rolle, ob die Pfade beim neutralen Element oder bei einem anderen Element beginnen.

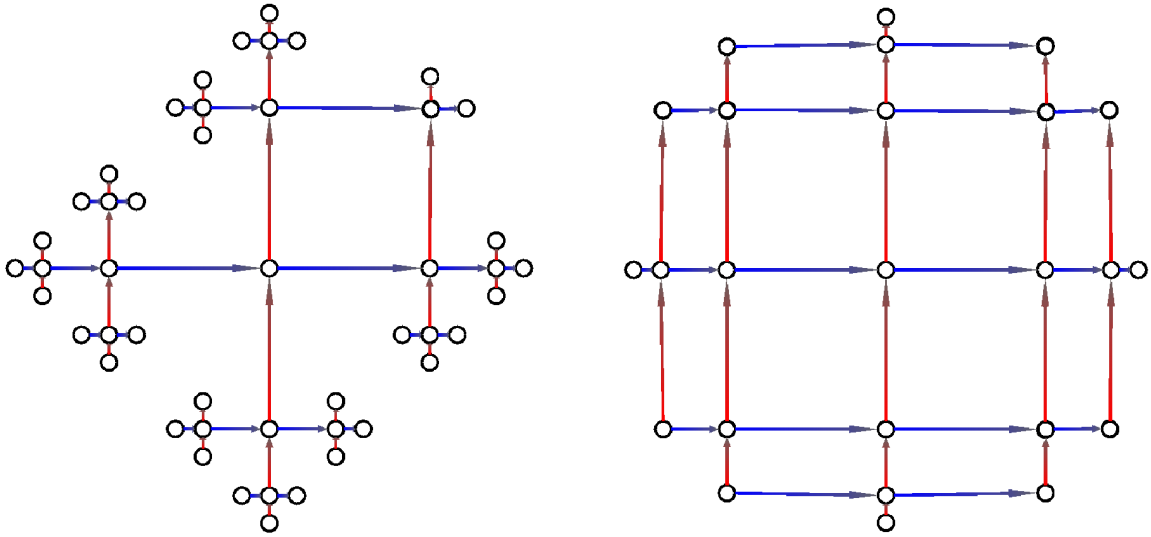
Bemerkung 1.1.8. *Eine Relation $w = v$ und eine Relation $wv^{-1} = \varepsilon$ sind gleich in dem Sinne, dass man sie austauschen kann, ohne die Gruppe zu verändern.*
Im Allgemeinen sind alle Relationen, die durch Links- oder Rechtsmultiplikation ineinander überführt werden können, gleich.

Wir haben die Menge aller Wörter, die zu einem bestimmten Wort w äquivalent sind noch nicht formal definiert. Um dies zu tun, führen wir das Konzept des Relators ein.

Definition 1.1.9. Hat eine Relation die Form $v = \varepsilon$, so schreiben wir nur $v \in F(S)$ als ein gekürztes Wort über dem Erzeugeralphabet. Wir nennen v einen *Relator*.

Definition 1.1.10. Sei R eine Menge von Relationen der Form $r_1 = r_2$. Diese Relationen lassen sich nach Bemerkung 1.1.8 in Relatoren $\tilde{R} \subseteq F(S)$ der Form $r_1 r_2^{-1} \in \tilde{R}$ umformen. Sei $w = w_1 ... w_n$ ein Wort aus Erzeugern der freien Gruppe $F(S)$. Eine *Äquivalenzumformung* ist:

- Das Einsetzen von $g^{-1}rg$ für $r \in \tilde{R}, g \in F(S)$ in w: $w_1 ... w_k g r g^{-1} w_{k+1} ... w_n$ mit anschließendem Kürzen
- Das Einsetzen von $g^{-1}r^{-1}g$ für $r \in \tilde{R}, g \in F(S)$ in w: $w_1 ... w_k g r g^{-1} w_{k+1} ... w_n$ mit anschließendem Kürzen



(a) Identifikation der Terme rb, br (b) Vollständige Anwendung von $rb = br$

Abbildung 1.2: Anwendung der Relation $rb = br$

Zwei Wörter v, w sind genau dann *äquivalent* (zum Symbol $\cong$) unter R bzw. $\tilde{R}$, wenn es eine Folge von Wörtern $v = w_1, ..., w_m = w$ gibt, die man konsekutiv durch eine Äquivalenzumformung ineinander überführen kann.

Es mag erst verwundern, dass wir hier $g^{-1}rg$ statt nur einem r einsetzen. Dadurch modellieren wir aber den Prozess "virtuelle Gruppenelemente" $g^{-1}g$ zu erzeugen, zwischen die dann r gesetzt wird. In der Definition wird weiterhin das Entfernen eines Relators r aus einem Wort arb nicht als eigenständige Äquivalenzumformung behandelt. Das ist auch nicht nötig, da wir den Effekt durch das Einsetzen des inversen Elements replizieren können: $arr^{-1}b$.

Satz 1.1.11. *Sei $R = \{r_1, ..., r_k\} \subseteq F_n$ eine Menge von Relatoren. Sei $[\varepsilon]$ die Menge aller Wörter, die äquivalent zum leeren Wort sind, d.h. durch Relationen in das leere Wort überführt werden können. Es gilt:*

$$[\varepsilon] = \langle R \rangle^{\triangleleft}_{F(S)}$$

Wobei $\langle R \rangle^{\triangleleft}_{F(S)}$ der kleinste Normalteiler der freien Gruppe $F(S)$ ist, der R enthält.

Beweis. Beobachte, dass $[\varepsilon]$ ein Normalteiler ist: Für ein Element $w \in [\varepsilon]$ gilt $g^{-1}wg \cong g^{-1}ww^{-1}g = \varepsilon$. Somit gilt $[\varepsilon] \supseteq \langle R \rangle^{\triangleleft}_{F(S)}$
Um die andere Richtung der Inklusion zu zeigen, generiere alle Elemente, die äquivalent zu ε sind. Wir zeigen durch Induktion, dass alle erzeugten Elemente in $\langle R \rangle^{\triangleleft}_{F(S)}$ liegen. Das neutrale Element ε und alle $r \in R$ liegen in $\langle R \rangle^{\triangleleft}_{F(S)}$. Zudem ist $g^{-1}rg \in \langle R \rangle^{\triangleleft}_{F(S)}$ für alle $g \in F(S), r \in R$.
Jedes andere Element aus $[\varepsilon]$ ist durch endlich viele Äquivalenzumformungen aus dem leeren Wort ε erzeugbar. Sei $w = ab \in \langle R \rangle^{\triangleleft}_{F(S)}$. Dann ist $a(g^{-1}rg)b = (ab)(g^{-1}rg) \in$

$\langle R\rangle^{\triangleleft}_{F(S)}$. Da jedes Element aus $[\varepsilon]$ induktiv durch Äquivalenzumformungen definiert ist, folgt: $[\varepsilon] \subseteq \langle R\rangle^{\triangleleft}_{F(S)}$. □

Wir können nun die durch Erzeuger und Relatoren erzeugte Gruppe formalisieren:

Definition 1.1.12. Eine *Präsentation einer Gruppe G* besteht aus einer Menge von Zeichen S (genannt *Generatoren*) und einer Menge von Wörtern $R \subseteq F(S)$ (genannt *Relatoren*). Wir bezeichnen mit $\langle S, R\rangle := F(S)/\langle R\rangle^{\triangleleft}_{F(S)}$ die aus der Präsentation erzeugte Gruppe.

In der Definition der Präsentation werden Relatoren und keine Relationen gefordert. Da jedoch Relationen generell leichter zu lesen sind und ohnehin in eindeutiger Weise in einen Relator umgewandelt werden können, werden wir beides austauschbar verwenden.

Definition 1.1.13. Lässt sich eine Gruppe als Präsentation mit endlich vielen Generatoren schreiben, bezeichnen wir sie als *endlich erzeugbar*.
Lässt sich eine Gruppe als Präsentation mit endlich vielen Generatoren und endlich vielen Relatoren schreiben, bezeichnen wir die Gruppe als *endlich präsentierbar*.

Beispiel 1.1.14. *Die einfachste Präsentation ist eine Präsentation ohne Relationen. Sei S eine endliche Menge von Zeichen. Dann ist*

$$\langle S \mid \emptyset\rangle = F(S)/\langle\emptyset\rangle^{\triangleleft}_{F(S)} = F(S)/\{e\} \cong F(S)$$

einfach die freie Gruppe.

Beispiel 1.1.15. *Betrachte die Gruppe* $G = \langle x, y \mid xyx^{-1} = y^2, yxy^{-1} = x^2\rangle$. *Die Gruppe ist trivial: Es gilt* $x = x(yx^{-1}xy^{-1}) = y^2xy^{-1} = yyxy^{-1} = yx^2$ *und somit* $x = y^{-1}$. *Demnach:* $y^{-2} = x^2 = yxy^{-1} = yy^{-1}y^{-1} = y^{-1}$. *Also* $x = y^{-1} = e$

Das letzte Beispiel zeigte eine Präsentationen, die überraschenderweise einfach die triviale Gruppe ist. Die Frage, ob ein bestimmtes Wort einer Präsentation einfach das neutrale Element ist, wird als das *Wortproblem* der Präsentation $\langle S \mid R\rangle$ bezeichnet. Zu zeigen, dass ein Gruppenelement trivial ist, ist nicht immer leicht und nicht mal immer möglich. Tatsächlich soll am Ende dieser Arbeit eine Gruppenpräsentation konstruiert werden, bei der es unmöglich ist festzustellen, ob bestimmte Produktterme zum neutralen Element evaluieren. Wie kann man ausschließen, dass ein Gruppenelement trivial ist?
Eine Möglichkeit ist, eine Normalform für die Gruppe zu finden.

Definition 1.1.16. Sei $G = \langle S, R\rangle$ eine Gruppe. Eine Normalform eines Gruppenelements $g \in G$ ist eine standardisierte, eindeutige Darstellung als Wort $w \in F(S)$ über einer Erzeugermenge S.

Normalformen erlauben uns, Äquivalenz und Nicht-Äquivalenz von zwei Wörtern $s_1...s_n$, $\tilde{s}_1...\tilde{s}_m \in F(S)$ zu zeigen. Die Wörter repräsentieren das gleiche Gruppenelement, wenn beide Wörter durch Anwendung von Relationen in die gleiche Normalform gebracht werden

können. Sie repräsentieren unterschiedliche Gruppenelemente, wenn die Wörter durch Anwendung von Relationen in zwei unterschiedliche Normalformen gebracht werden können. Bekannte Beispiele für Normalformen außerhalb der Gruppentheorie sind die eindeutige Darstellung eines Bruches als vollständig gekürzte Zahl, die Chomsky-Normalform einer Grammatik oder die Jordansche Normalform einer Matrix.

Satz 1.1.17. *Seien $G = \langle S_G, R_G \rangle$ und H zwei Gruppen. Betrachte die Abbildung $\varphi : S_G \to H$. Definiere die Fortsetzung $\tilde{\varphi} : G \to H$ folgendermaßen: $\tilde{\varphi}(w) = \tilde{\varphi}(s_1...s_n) = \tilde{\varphi}(s_1)...\tilde{\varphi}(s_n) \in H$ für eine beliebige Darstellung von $w \in G$ als Wort $s_1...s_n \in F(S_G)$.*
$\tilde{\varphi}$ ist ein wohldefinierter Homomorphismus $\tilde{\varphi} : G \to H$, wenn $\tilde{\varphi}$ die Relationen R_G von G achtet, d.h. Für jeden Relator $s_1...s_n \in R_G$ gilt $\varphi(s_1)...\varphi(s_n) = e$.

Beweis. $\tilde{\varphi}$ ist wohldefiniert, wenn alle Darstellungen von w auf das gleiche Element in H abbilden.
Dies ist hier erfüllt. Zwei Darstellungen $s_1...s_n, s'_1...s'_n \in F(S_G)$ evaluieren als das gleiche Gruppenelement, wenn sie sich durch Äquivalenzübergänge ineinander überführen lassen. Die Äquivalenzübergänge entstehen durch Einsetzen von $g^{-1}rg$ oder $g^{-1}r^{-1}g$ für $r \in R_G, g \in F(S_G)$. Da $\tilde{\varphi}$ Relationen achtet, ist $\tilde{\varphi}(grg^{-1}) = \tilde{\varphi}(gg^{-1}) = e$ und $\tilde{\varphi}(gr^{-1}g^{-1}) = \tilde{\varphi}(gg^{-1}) = e$. Das Bild $\tilde{\varphi}(s_1...s_n)$ bleibt also nach Anwendung von Äquivalenzübergängen gleich. Durch Induktion über die benötigten Umformungsschritte zwischen w und w', ist $\tilde{\varphi}$ somit wohldefiniert.
Die Homomorphismuseigenschaft von $\tilde{\varphi}$ folgt direkt aus der Definition. □

Satz 1.1.18. *Jede Gruppe G resultiert aus einer Präsentation:*

$$G \cong \langle S \mid R \rangle = \langle \{g \in G\} \mid \{gh = k : \forall g, h, k \in G, gh = k\} \rangle$$

Wir kürzen diese Präsentation durch $\langle G \mid \emptyset \rangle$ ab, indem wir die Gruppe zu den Generatoren schreiben. Im Normalfall ist dabei klar, dass nicht die freie Gruppe über dem Alphabet G gemeint ist.

Beweis. Es gibt einen natürlichen Homomorphismus $\varphi : F(S) \to G$, der dadurch definiert ist, dass wir jeden Generator S auf das korrespondierende Gruppenelement G abbilden.
Wir beobachten, dass für ein Wort $w \in F(S)$: $\varphi(w) = e$ genau dann, wenn das schrittweise Ausmultiplizieren von jeweils zwei Zeichen nach den Regeln $\{gh = k : \forall g, h, k \in G, gh = k\}$ im Element e endet.
G besitzt damit die Präsentation $\langle \{g \in G\} \mid \{gh = k : \forall g, h, k \in G, gh = k\} \rangle$. □

1.2 HNN-Erweiterungen

HNN-Erweiterungen, benannt nach Graham Higmann, Bernhard Neumann und Hanna Neumann sind ein nützliches Hilfsmittel in der Gruppentheorie. HNN-Erweiterung werden genutzt, um zwei Untergruppen durch das Einführen eines neuen Generators zu konjugieren. Wir werden auf diese zurückgreifen, um einen Konfigurationsübergang einer Turingmaschine als Konjugationsrelation zweier Untergruppen zu kodieren.

Definition 1.2.1. Sei G eine Gruppe und seien $A, B \subseteq G$ zwei Untergruppen mit einem Isomorphismus $\varphi : A \to B$. Wir konstruieren die *HNN-Erweiterung über der Gruppe G* als eine neue Gruppe, die die Relationen von G enthält. Außerdem fügen wir einen neuen Generator t und neue Relationen hinzu, mit der A und B zu konjugierten Untergruppen werden:

$$G*_A = \left\langle G \sqcup \{t\} \mid \{t^{-1}at = \varphi(a) : \forall a \in A\} \right\rangle$$

Das Zeichen t bezeichnet man als das *stabile Zeichen.*

Beispiel 1.2.2. *Wir beginnen mit einem trivialen Beispiel. Sei* $G = \{e\} = \langle \emptyset \mid \emptyset \rangle$ *trivial und* $A = B = \{e\}$ *sowie der Isomorphismus* $\varphi = id : A \to B$. *Dann erhalten wir die HNN-Erweiterung*

$$G*_{\{e\}} = \left\langle \{t\} \mid t^{-1}et = e \right\rangle = \langle \{t\} \mid \emptyset \rangle \cong \mathbb{Z}$$

Beispiel 1.2.3. *Ein zweites simples Beispiel. Sei* $G = F_n \cong \langle \{s_1, ..., s_n\} \mid \emptyset \rangle$ *und* $A = B = \{e\}$ *mit* $\varphi = id$. *Dann gilt*

$$G*_{\{e\}} = \langle s_1, ..., s_n, t \mid \emptyset \rangle \cong F(\{s_1, ..., s_n, t\}) \cong F_{n+1}$$

Beispiel 1.2.4. *Sei* $V_4 = \langle a, b \mid aa, bb, abab \rangle$ *die Kleinsche Vierergruppe. Wir konjugieren die Untergruppen* $A = \{e, a\}$ *und* $B = \{e, b\}$ *mit* $\varphi(e) = e, \varphi(a) = b$. *Wir erhalten:*

$$G*_A = \left\langle a, b, t \mid aa, bb, abab, t^{-1}at = b \right\rangle$$

Beispiel 1.2.5. *Betrachte wieder die Kleinsche Vierergruppe* $V_4 = \langle a, b \mid aa, bb, abab \rangle$. *Wir konjugieren nun die Untergruppen* $A = B = \{e, a\}$ *mit* $\varphi = id$. *Wir erhalten:*

$$G*_A = \left\langle a, b, t \mid aa, bb, abab, t^{-1}at = a \right\rangle$$

Wir wollen beweisen, dass die letzten beiden Präsentationen unterschiedliche, d.h. nicht isomorphe Gruppen definieren.
Zu zeigen, dass zwei verschiedene Präsentationen isomorphe Gruppen erzeugen, ist allerdings im Allgemeinen für endlich erzeugte Gruppen kein lösbares und schon gar kein einfaches Problem. (Siehe dazu [BGBGL08]). Will man zeigen, dass zwei Gruppen unterschiedlich sind, besteht daher der erste Schritt meistens darin, die beiden Gruppen zu vereinfachen.
Das tun wir im Folgendenen durch die Abelisierung. Wir fügen der Gruppe dazu Relationen hinzu, sodass die beiden Gruppen kommutativ werden. Dann zeigen wir, dass die Abelisierungen der beiden Gruppen verschieden sind.

Definition 1.2.6. Sei G eine Gruppe. Dann bezeichnen wir mit

$$G^{ab} = \langle G \mid \{gh = hg : g, h \in G\} \rangle$$

die *Abelisierung* der Gruppe G.

Beispiel 1.2.7. *Sei $G = V_4*_A$ die HNN-Erweiterung von Beispiel 1.2.4 und $H = V_4*_A$ die HNN-Erweiterung von Beispiel 1.2.5.*
Die beiden Gruppen sind nicht isomorph.

Beweis. Sei G^{ab}, H^{ab} die Abelisierung on G und H. Da jedes Gruppenelement als Produkt der Erzeuger a, b, t geschrieben werden kann reicht es bereits aus, die Kommutativitätsrelationen für die Erzeuger hinzuzufügen. Es gilt also

$$G^{ab} \cong \langle a, b, t : aa, bb, abab, t^{-1}at = b, ab = ba, at = ta, bt = tb \rangle$$

$$H^{ab} \cong \langle a, b, t : aa, bb, abab, t^{-1}at = a, ab = ba, at = ta, bt = tb \rangle$$

Entfernt man redundante Relationen erhält man

$$\begin{aligned} G^{ab} &\cong \langle a, b, t : aa, bb, t^{-1}at = b, ab = ba, at = ta, bt = tb \rangle \\ &\cong \langle a, b, t : aa, bb, t^{-1}ta = b, ab = ba, at = ta, bt = tb \rangle \\ &\cong \langle a, t : aa, at = ta \rangle \cong C_2 \times \mathbb{Z} \\ H^{ab} &= \langle a, b, t : aa, bb, ab = ba, at = ta, bt = tb \rangle \cong C_2 \times C_2 \times \mathbb{Z} \end{aligned}$$

welche offensichtlich nicht isomorph sind. □

Die Unterschiede von G und H sind ebenfalls gut an ihren Cayley-Graphen zu erkennen. In der folgenden Abbildung stehen zweifach gezeichnete Kanten für den Generator a, einfach gezeichnete Kanten für b. Die Pfeile nach oben stehen für die Multiplikation mit dem freien Zeichen t. Linksnebenklassen von V_4 sind grau markiert.
Die Bilder sind vollkommen anders, der verwendete Isomorphismus φ spielt also eine große Rolle!
Die Abbildung zeigt, dass die Gruppe V_4 Teil der HNN-Erweiterung V_4*_A ist. Es stellt sich die Frage, ob dieser Sachverhalt für beliebige HNN-Erweiterungen gilt.

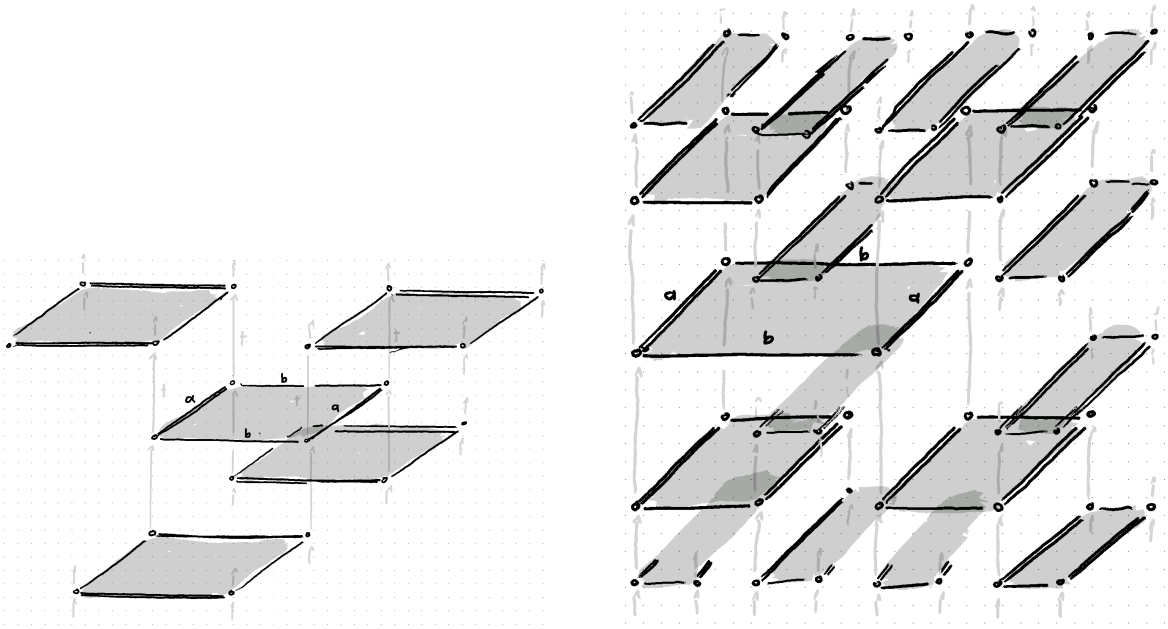
Satz 1.2.8 (Brittons Lemma)**.** *Sei G eine Gruppe, $A, B \subseteq G$ Untergruppen mit Isomorphismus $\varphi : A \to B$, sodass $H = G*_A$ eine HNN-Erweiterung zum stabilen Zeichen t ist. Dann gilt $G \subseteq G*_A$.*
Wählen wir Repräsentantensysteme R_A von G/A und R_B von G/B (wobei wir für die Nebenklassen A, B jeweils das neutrale Element e wählen), so lässt sich jedes nichttriviale Element $h \in G*_A$ in eindeutiger Weise als Produkt schreiben:

$$h = r_1 t^{\varepsilon_1} r_2 t^{\varepsilon_2} ... r_n t^{\varepsilon_n} g_{n+1}$$

mit $\varepsilon_i = \pm 1, g_{n+1} \in G, nin\mathbb{N}_0$ und mit $g_i \in R_A$, wenn $\varepsilon_i = +1$ und $g_i \in R_B$, wenn $\varepsilon = -1$. Außerdem enthält der obere Ausdruck kein Teilwort $t^{-1}et$ oder tet^{-1}.

Beweis. Siehe Theorem 1.7 aus [SW79]. □

Es ist möglich, mehrere HNN-Erweiterungen gleichzeitig auf eine Gruppe anzuwenden. Dies machen wir, indem wir mehrere stabile Zeichen und Relationen hinzufügen.



(a) $V_4 *_A$ wie in **Beispiel 1.2.3** (b) $V_4 *_A$ wie in **Beispiel 1.2.4**

Abbildung 1.3: Cayley-Graphen der Gruppen $V_4 *_A$

Definition 1.2.9. Sei G eine Gruppe und seien $\varphi_i : A_i \to B_i$ Isomorphismen zwischen zwei Familien von Untergruppen. Dann ist die Gruppe

$$G*_{\{H_i:i\in I\}} := \left\langle G \sqcup \{t_i : i \in I\} \mid \{t_i^{-1} a t_i = \varphi_i(a) : a \in A\} \right\rangle$$

eine *(mehrfache) HNN-Erweiterung* zu den stabilen Zeichen t_i für $i \in I$.

Satz 1.2.10. *Sei G eine Gruppe und seien $\varphi_i : A_i \to B_i$ mit $i \in \{1, ..., n\}$ Isomorphismen zwischen Untergruppen, sodass $G*_{\{H_i:i\in\{0,...,n\}\}}$ eine mehrfache HNN-Erweiterung ist. Dann gilt $G \subseteq G*_{\{H_i:i\in I\}}$*

Beweis. Die HNN-Erweiterung $G*_{\{H_i:i\in I\}}$ entsteht durch Hinzufügen von stabilen Zeichen und Relatoren. Anstelle alle gleichzeitig hinzuzufügen, kann man sie auch nacheinander einbauen. Man erhält dadurch die Gleichung:

$$G*_{\{H_i:i\in\{1,...,n\}\}} = G *_{H_1} ...*_{H_n}$$

Somit folgt $G \subseteq G*_{\{H_i:i\in I\}}$ aus Induktion. □

Im letzten Beweis charakterisierten wir HNN-Erweiterungen als das Hinzufügen von stabilen Zeichen t und Relationen. Mit dieser Charakterisierung lassen sich HNN-Erweiterungen auch bei Untergruppen durchführen.

Definition 1.2.11. Sei G eine Gruppe und $\phi : A_i \to B_i$ Isomorphismen zwischen zwei Familien von Untergruppen, sodass $G*_{\{A_i:i\in I\}}$ eine HNN-Erweiterung ist.
Wir definieren mit

$$U*_{\{A_i:i\in I\}} = \left\langle U \sqcup \{t_i : i \in I\} \mid \{t_i^{-1} a t_i = \phi_i(a) : a \in A \cap U\}\right\rangle$$

die *(Pseudo-)HNN-Erweiterung der Untergruppe* $U \subseteq G$.

Definition 1.2.12. Sei G eine Gruppe und $\phi : A_i \to B_i$ Isomorphismen zwischen zwei Familien von Untergruppen, sodass $G*_{\{A_i:i\in I\}}$ eine HNN-Erweiterung ist.
Eine Untergruppe $U \subseteq G$ wird *gut* genannt, wenn $\phi_i(U \cap A_i) = U \cap B_i$ für alle $i \in I$.
In dem Fall ist

$$U*_{\{A_i:i\in I\}} = \left\langle U \sqcup \{t_i : i \in I\} \mid \{t_i^{-1} a t_i = \phi_i(b) : a \in A \cap U\}\right\rangle$$

eine *echte HNN-Erweiterung der Untergruppe* $U \subseteq G$ zu den stabilen Zeichen t_i und den Isomorphismen $\phi|_U : A_i \cap U \to B_i \cap U$

Korollar 1.2.13. *Sei G eine Gruppe und $\varphi_i : A_i \to B_i$ mit $i \in \{1, ..., n\}$ Isomorphismen zwischen Untergruppen, sodass $G*_{\{A_i:i\in I\}}$ eine HNN-Erweiterung ist. Sei U eine gute Gruppe und $U*_{\{A_i:i\in I\}}$ eine HNN-Erweiterung der Untergruppe.*
Dann gilt Brittons Lemma auch für $U*_{\{A_i:i\in I\}}$ und es gilt $U *_{\{A_i:i\in I\}} \cap G = U$.

Beweis. Die HNN-Erweiterung der Untergruppe ist einfach eine HNN-Erweiterung zu den isomorphen Untergruppen $U \cap A_i \cong U \cap B_i$. Somit gilt Brittons Lemma.
Es folgt: $U \subseteq U*_{\{A_i:i\in I\}}$ und damit $U *_{\{A_i:i\in I\}} \cap G = U$. □

1.3 Graphen und Bäume

Bass-Serre-Theorie untersucht Wirkungen von Gruppen auf Bäumen. Hierfür sollen nun Bäume und Graphen in Anlehnung an [Ser02] definiert werden.

Definition 1.3.1. Ein *Graph* ist ein Tupel $\Gamma = (X, Y)$ bestehend aus einer Menge von *Knoten* X und einer Menge von *Kanten* Y. Jede Kante hat einen Start- und einen Endknoten, definiert durch Abbildungen o (Origin) und t (Terminus)

$$\begin{aligned} o : Y &\to X & \qquad t : Y &\to X \\ y &\mapsto o(y) & \qquad y &\mapsto t(y) \end{aligned}$$

Für jede Kante $\bar{y} \in Y$ existiert eine *inverse Kante* $\bar{y} \in Y$ für die gilt: $o(\bar{y}) = t(y)$ und $t(\bar{y}) = o(y)$.

Da inverse Kanten immer ebenfalls Element der Kantenmenge sind und da zwischen zwei Knoten auch mehrere Kanten möglich sind, würde man den definierten Graphen in der Informatik als einen ungerichteten Multigraphen bezeichnen.
Die Graphen müssen nicht endlich sein. Später werden wir Berechnungen auf unendlichen Graphen durchführen.

Abbildung 1.4: Graph in Form eines Segments

Beispiel 1.3.2. *Ein Segment ist ein Graph, bestehend aus zwei Knoten, die durch eine Kante verbunden sind.*

Beispiel 1.3.3. *Eine Schleife ist ein Graph, bestehend aus einem Knoten, der durch eine Kante mit sich selbst verbunden ist.*

Abbildung 1.5: Graphen in Form einer Schleife und einer Doppelschleife

Definition 1.3.4. Die *Valenz eines Knotens* p gibt an, wie viele ausgehende Kanten der Knoten hat. Besitzen alle Knoten p die gleiche Valenz, so ist die *Valenz des Graphen* wohldefiniert und gleich der Valenz der Knoten.

Definition 1.3.5. Ein *Pfad von Länge* n in einem Graphen $\Gamma = (X, Y)$ (für $n \geq 1$) ist eine Folge $(y_1, ..., y_n)$ von Kanten $y_i \in Y$ mit $t(y_i) = o(y_{i+1})$.
Unter einem Pfad der Länge 0 verstehen wir eine leere Folge von Kanten (), beginnend (und endend) bei einem Knoten $x \in X$.
Wird keine Kante direkt zurückgelaufen, d.h. $y_i \neq \bar{y}_{i+1}$ für $i \in \{0, ..., n-1\}$, so sagen wir, der Pfad enthält *keine Rückverfolgung*.

Die Definition eines Pfades ohne Rückverfolgung schließt nicht aus, dass ein Knoten über einen längeren Umweg doppelt besucht wird.

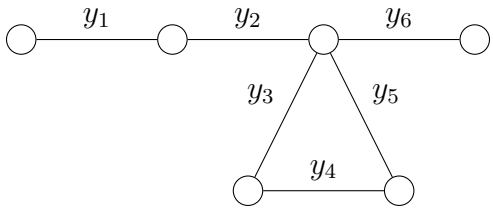


Abbildung 1.6: Pfad, der einen Knoten mehrfach besucht

Definition 1.3.6. Ein Graph wird als *zusammenhängend* bezeichnet, wenn es für alle zwei Knoten $x, \tilde{x} \in X$ einen Pfad $(y_1, ..., y_n)$ zwischen den beiden Knoten gibt, sodass $o(y_1) = x, t(y_n) = \tilde{x}$

Definition 1.3.7. Ein *Zykel* ist ein kreisförmiger Pfad $(y_1, ..., y_n)$ der Länge $n \geq 1$. Wir fordern: $o(y_1) = t(y_n)$.
Wie bei Pfaden sagen wir, ein Zykel enthält *keine Zurückverfolgung*, wenn $(y_1, ..., y_n)$ ein Pfad ohne Zurückverfolgung ist und $y_1 \neq \bar{y}_n$ gilt.

Definition 1.3.8. Ein Graph wird ein *Baum* genannt, wenn er zusammenhängend ist und keinen Zykel ohne Zurückverfolgung enthält.

Lemma 1.3.9. *Sei $\Gamma = (X, Y)$ ein Graph. Gibt es zwei verschiedene Knoten $x, x' \in X$, die durch zwei verschiedene Pfade ohne Zurückverfolgung $(y_1, ..., y_n), (\tilde{y}_1, ..., \tilde{y}_m)$ miteinander verbunden sind, d.h.*

- $o(y_1) = o(\tilde{y}_1) = x, t(y_n) = t(\tilde{y}_m) = x'$
- $n \neq m$ *oder* $y_i \neq \tilde{y}_i$ *für ein* $i \in \{0, ..., n\}$

dann enthält Γ einen Zykel ohne Zurückverfolgung.

Beweis. o.E. $n \leq m$. Zeige den Satz durch Induktion über die Länge des Pfades $n \geq 0$.
Falls $n = 0, m \geq 1$, dann ist $(\tilde{y}_1, ..., \tilde{y}_m)$ bereits ein Zykel. Handelt es sich um einen Zykel mit Zurückverfolgung, so ist zwangsweise $y_1 = \bar{y}_n$. In dem Falle ist $(\tilde{y}_2, ..., \tilde{y}_{m-1})$ ein kürzerer Zykel. Durch einen Induktionsbeweis lässt sich zeigen, dass man durch wiederholtes Anwenden dieser Kürzung einen Zykel ohne Zurückverfolgung erhält.
Falls $n \geq 1, m \geq 1$, seien $(y_1, ..., y_n)$ und $(\tilde{y}_1, ..., \tilde{y}_m)$ zwei verschiedene Pfade mit Länge $n, m \in \mathbb{N}_0$. Sei o.E. $y_1 \neq \tilde{y}_1$. Wäre $y_1 = \tilde{y}_1$, dann wären $(y_2, ..., y_n)$ und $(\tilde{y}_2, ..., \tilde{y}_m)$ zwei verschiedene, kürzere Pfade beginnend am Knoten $o(y_2) = o(\tilde{y}_2)$ und die Anwendung der Induktionsvoraussetzung gibt den gesuchten Zykel.
Dann ist $(\bar{y}_n, ..., \bar{y}_1, \tilde{y}_1, ..., \tilde{y}_m)$ ein Zykel mit Anfang und Ende bei x'. Handelt es sich auch hier um einen Zykel mit Rückverfolgung folgt daraus $\tilde{y}_m = \bar{y}_n$. Wie eben erhält man mit $(\bar{y}_{n-1}, ..., \bar{y}_1, \tilde{y}_1, ..., \tilde{y}_{m-1})$ einen kürzeren Zykel. Mit Induktion kann man dann einen Zykel ohne Rückverfolgung finden. □

Korollar 1.3.10. *Sei $T = (X, Y)$ ein Baum. Für zwei Knoten $x, x' \in X$ existiert ein eindeutiger Pfad ohne Zurückverfolgung von x nach x'.*

Beweis. Der Pfad existiert, da der Baum zusammenhängend ist. Der Pfad ist eindeutig, da der Baum sonst einen Zykel enthalten würde. □

Definition 1.3.11. Ein *Graphautomorphismus* von $\Gamma = (X, Y)$ ist eine bijektive Abbildung der Knoten und Kanten $\varphi : X \bigsqcup Y \to X \bigsqcup Y$, die die Graphstruktur erhält. Das bedeutet, der Automorphismus:

- erhält Kanten und Knoten: $\forall y \in Y : \varphi(y) \in Y, \forall x \in X : \varphi(x) \in X$
- erhält inverse Kanten: $\forall y \in Y : \varphi(\bar{y}) = \varphi(\bar{y})$
- erhält Startknoten: $\forall y \in Y : o(\varphi(y)) = \varphi(o(y))$

- erhält Endknoten: $\forall y \in Y : t(\varphi(y)) = \varphi(t(y))$

Die Menge aller Automorphismen eines Graphen bilden eine Gruppe. Die Gruppe wird mit $Aut(\Gamma)$ notiert.

Die Elemente von $Aut(\Gamma)$ beschreiben Operationen, die man auf den Graphen anwenden kann, die den Graphen gleich lassen. $Aut(\Gamma)$ beschreibt somit alle Symmetrien des Graphen Γ.

Beispiel 1.3.12. *Betrachte Abbildung 1.7. Diese zeigt einen Graphen in Form eines "+"-Symbols. Die Rotation aller Kanten und Knoten um dem mittleren Knoten ist ein Automorphismus, da die Abbildung den Graphen erhält.*

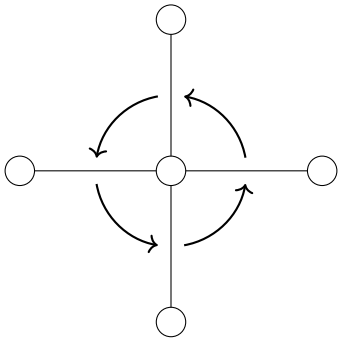


Abbildung 1.7: Graph mit Automorphismus

Definition 1.3.13. Eine Gruppe G *wirkt durch Gruppenoperation* auf einem Graphen $\Gamma = (X, Y)$, wenn es einen Gruppenhomomorphismus $\varphi : G \to Aut(\Gamma)$ gibt.
Diese Gruppenoperation ist eine Gruppenoperation auf einer Menge X im herkömmlichen Sinne, erlaubt also dessen Untersuchung durch Orbits und Stabilisatoren.
Statt $\varphi(g)(\Gamma)$ schreiben wir, wie bei Gruppenoperationen üblich, $g\Gamma$.
Statt $\varphi(g)(p)$ für einen Knoten $p \in X$ schreiben wir gp.
Statt $\varphi(g)(y)$ für eine Kante $y \in Y$ schreiben wir gy.

Es wird davon ausgegangen, dass dem Leser Gruppenoperationen bekannt sind. Orbits, Stabilisatoren und transitive Wirkungen werden nicht mehr definiert.

Korollar 1.3.14. *Eine Folgerung aus der Definition für Gruppenoperationen als Gruppenhomomorphismen ist, dass $\varphi : G \to Aut(\Gamma)$ durch die Bilder der Erzeuger von G festgelegt ist.*
Wir werden das im Folgenden nutzen, um übersichtlich Beispiele von Gruppenoperationen anzugeben.

Beispiel 1.3.15. *Die 4-elementige zyklische Gruppe $C_4 := \langle g \mid g^4 \rangle$ operiert auf dem Graphen aus Abbildung 1.7, indem man das Gruppenelement g mit der Rotation um $90°$ identifiziert.*

Beispiel 1.3.16. *Die 8-elementige zyklische Gruppe $C_8 := \langle g \mid g^8 \rangle$ operiert auf dem Graphen aus Abbildung 1.7, indem das Gruppenelement g als $90°$ Rotation um den mittleren Knoten wirkt.*
g^4 beschreibt dann eine Volldrehung oder einfacher die Identität und g^5 ist wieder die $90°$ Rotation. g und g^5 haben also die gleiche Wirkung auf dem Graphen!

Definition 1.3.17. Vertauscht eine Gruppenoperation auf einem Graphen den Anfangs- und Endknoten einer Kante, so sagen wir, dass die Gruppenoperation die Kante *invertiert*. Eine Gruppenoperation, die keine Kante invertiert, wird als *Gruppenoperation ohne Inversion* bezeichnet.

Beispiel 1.3.18. *Die 2-elementige zyklische Gruppe $C_2 := \langle g \mid g^2 \rangle$ operiert auf dem segmentförmigen Graphen Γ durch Spiegelung an der Kante. Dann operiert das Gruppenelement g mit Inversion auf Γ.*

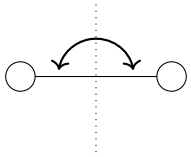


Abbildung 1.8: Gruppenoperation mit Inversion

Kapitel 2

Bass-Serre Theorie von HNN-Erweiterungen

Im Buch Trees ([Ser02]) stellt Jean-Pierre Serre eine Methode vor, Gruppen zu untersuchen. Er nimmt an, eine Gruppe würde auf einem Baum operieren. Durch den Quotientengraphen erhält er Aufschluss über die Struktur der Gruppe. Auf diese Weise kann man beispielsweise feststellen, ob eine Gruppe ein amalgamiertes Produkt zweier Gruppen ist. Wir werden die im Buch beschriebenen Methoden nutzen, um zu erkennen ob eine Gruppe eine HNN-Erweiterung ist und wie sie auf Bäumen operiert. Dann bauen wir eine Äquivalenz zwischen Pfaden ohne Zurückverfolgung im Baum und der Normalform für HNN-Erweiterungen nach Britton auf.

2.1 Quotientengraphen

Definition 2.1.1. Sei $\Gamma = (X, Y)$ ein Graph und G eine Gruppe, die auf Γ operiert. Wir definieren einen Quotientengraphen durch $\Gamma/G := (X/G, Y/G) := (X/\sim, Y/\sim)$ mit $x_1 \sim x_2 \iff x_2 = g \cdot x_1$ für ein $g \in G$ und $y_1 \sim y_2 \iff y_2 = g \cdot y_1$ für ein $g \in G$.
Die Start und Endpunkte einer Kante Gy sind definiert durch $o(Gy) = Go(y), t(Gy) = Gt(y)$.

Beweis. Um die Wohldefiniertheit sicherzustellen, ist zu zeigen, ob die Definition Start- und Endpunkte bei Kanten erhält. Sei $y' \in Gy$. Dann gibt es ein $g \in G$, sodass $gy = y'$. Dann gilt $go(y) = o(gy) = o(y'), gt(y) = t(gy) = t(y')$ und damit $o(Gy) = Go(y)$ sowie $t(Gy) = Gt(y)$.
Nach unserer Definition für Graphen müssten wir noch zeigen, dass der Quotientengraph ungerichtet ist. Dazu nehmen wir zusätzlich an, die Gruppenwirkung operiert ohne Inversion. Jetzt gilt für jede Kante $y \in Y$: $Gy \neq G\bar{y}$. Damit gibt es für jedes Gy ein inverses $G\bar{y}$. □

Bei der Definition des Quotientengrahen fügten wir während des Wohldefiniertheitsbeweises die Bedingung hinzu, dass die Gruppenwirkung ohne Inversion operiert. Da Grup-

penwirkungen ab sofort immer ohne Inversion auftauchen werden, wird diese Annahme keine weiteren Probleme verursachen.

Beispiel 2.1.2. *Wir betrachten wieder den Graphen in Form des "+"-Symbols, auf dem die zyklische Gruppe C_4 operiert. (Abb. 2.1) Durch Rotation lassen sich die äußeren, roten Knoten aufeinander abbilden. Der innere, blaue Knoten bleibt unter der Gruppenwirkung fix. Der Quotientengraph hat somit die Form eines Segmentes.*

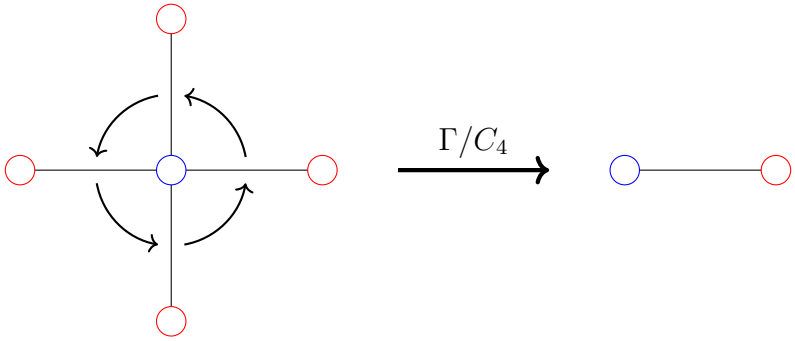


Abbildung 2.1: Segmentförmiger Quotient eines Graphen

Beispiel 2.1.3. *Wir betrachen einen Graphen Γ in Form einer unendlich langen Kette, wie in Abbildung 2.2 dargestellt. Auf dem Graph wirkt die Gruppe $(\mathbb{Z}, +)$ durch Verschiebung. Das Element $+1$ verschiebt hierbei die Knoten um eins nach rechts.*
Alle Knoten des Graphen liegen in einem Orbit und alle ungerichteten Kanten des Graphen liegen in einem Orbit. So erhält man einen Quotientengraphen in Form einer Schleife.

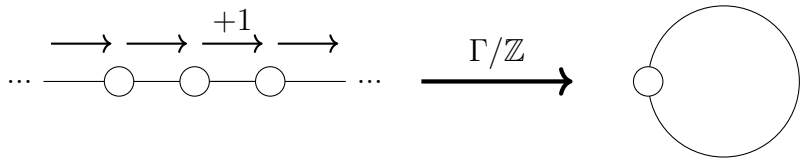


Abbildung 2.2: Schleifenförmiger Quotient eines Graphen

Die Gruppe $(\mathbb{Z}, +)$, die im letzten Beispiel verwendet wurde, ist isomorph zu der trivialen HNN-Erweiterung $\{e\}*_{\{e\}} = \langle t \mid \emptyset \rangle \cong \mathbb{Z}$. Dass als Quotient eine Schleife herauskommt, ist kein Zufall. Den Zusammenhang zwischen HNN-Erweiterungen und Schleifen untersuchen wir im nächsten Abschnitt.

2.2 Charakterisierung von HNN-Erweiterungen nach Bass-Serre

In diesem Abschnitt charakterisieren wir HNN-Erweiterungen als Gruppen, die mit einem besonderen Quotienten auf unendlichen Bäumen operieren. Dies soll uns helfen, HNN-

Erweiterungen besser zu verstehen. Wir erhalten das Resultat, dass ein Gruppe G genau dann auf einem Baum mit einer Schleife als Quotienten und ohne Inversion operiert, wenn G eine HNN-Erweiterung ist.
Das wollen wir zuerst an einem komplexeren Beispiel illustrieren.

Beispiel 2.2.1. *Sei $V_4 = \langle v, h \mid vh = hv, v^2, h^2 \rangle$ die Kleinsche Vierergruppe. Wir bilden die HNN-Erweiterung*

$$V_4 *_{\langle h \rangle} = \langle v, h, s \mid vh = hv, v^2, h^2, s^{-1}vs = h \rangle$$

an den Untergruppen $\langle g \rangle$ und $\langle h \rangle$. $V_4 *_{\langle h \rangle}$ wirkt auf dem 4-valenten Baum.
v und h wirken durch Vertauschen von jeweils zwei Ästepaaren. (Siehe Abbildungen 2.3 und 2.4). Die Vertauschung ist so, dass die Reihenfolge von Kanten innerhalb eines Knotens der getauschten Äste gleich bleibt. Dreht man sich auf einem Knoten eines vertauschten Astes gegen den Uhrzeigersinn sieht man eine rote Kante weggehen, eine blaue Kante weggehen, eine blaue Kante auf sich zukommen, eine rote Kante auf sich zukommen.
t wirkt auf dem Baum, indem Knoten entlang der mittleren blauen Pfeilkette verschoben werden. Äste werden dabei mitgezogen.

Alle Relationen $vh = hv, v^2, h^2, s^{-1}vs = h$ manifestieren sich auch in der Gruppenwirkung. Beispielsweise ist die Gruppenwirkung hv die gleiche wie vh.
Man kann zeigen, dass durch wiederholtes Anwenden der Operationen v, h, t alle Knoten auf dem mittleren Knoten und alle Kanten auf die Kante rechts vom mittleren Knoten abgebildet werden können. Der Quotient der Operation ist somit eine Schleife. Man kann zeigen, dass die Gruppe ohne Inversion operiert.

Wir werden die Behauptungen des letzten Beispiels nicht beweisen. Stattdessen beginnen wir mit einem einfachen Satz, der uns ein Gefühl für kommende Argumentationen geben soll.

Satz 2.2.2. *Sei $T = (X, Y)$ ein Baum, auf dem eine Gruppe G ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.*
Dann muss jeder Knoten von T die gleiche Valenz besitzen. Die Valenz ist ≥ 2.

Beweis. Da T/G als Schleife eine Kante besitzt, besitzt auch T mindestens eine Kante y. An dieser Kante hängen zwei unterschiedliche Knoten $p := o(y), q := t(y)$. T kann kein Segment bestehend aus p, q, y sein, da der einzige nicht-triviale Automorphismus die Vertauschung der beiden Knoten ist, dies jedoch eine Gruppenoperation mit Inversion wäre. T enthält daher mindestens einen dritten Knoten, der o.E. durch eine Kante mit p verbunden ist. p hat damit Valenz ≥ 2.
Da G transitiv auf T wirkt, gibt es für alle $x \in X$ ein $g \in G$ sodass $gp = x$. Automorphismen erhalten End- und Startpunkte von Kanten und somit muss x die gleiche Valenz wie p besitzen. □

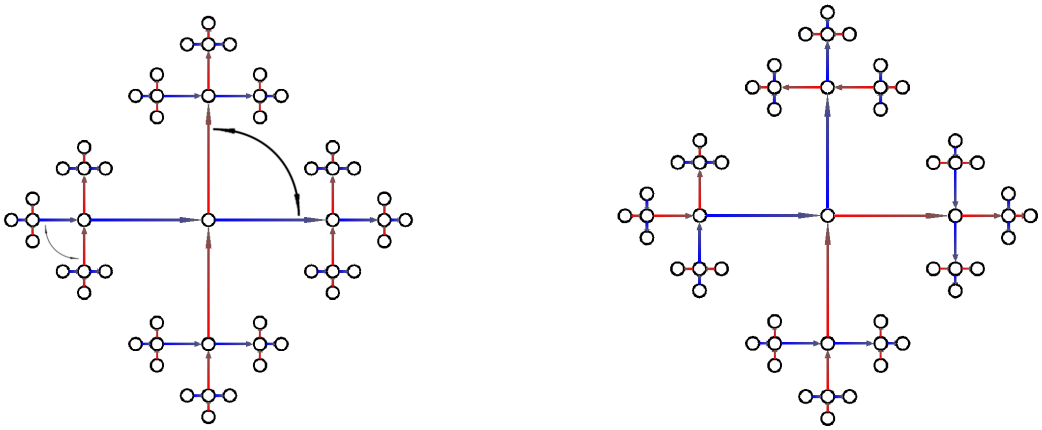


Abbildung 2.3: Anwendung der Gruppenwirkung h

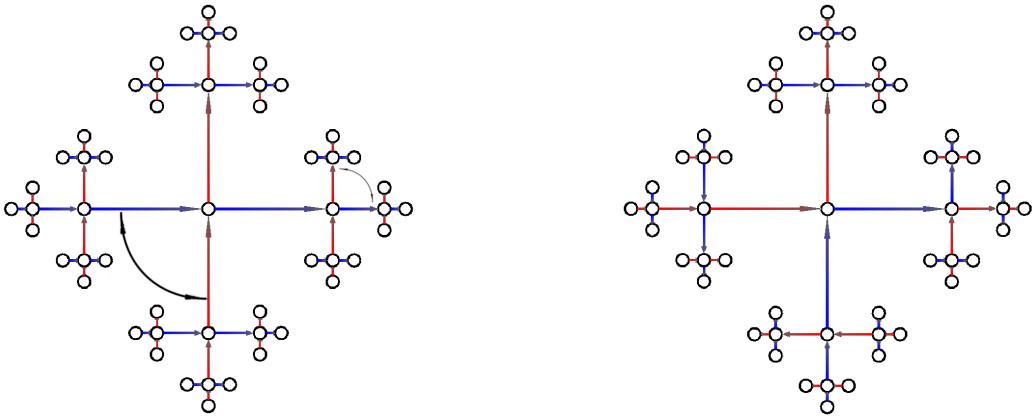


Abbildung 2.4: Anwendung der Gruppenwirkung v

Lemma 2.2.3. *Sei $\Gamma = (X, Y)$ ein Graph, auf dem eine Gruppe G wirkt.*
Dann erhält jede Gruppenwirkung $g \in G$ die Länge von Pfaden ohne Zurückverfolgung.

Beweis. Sei $(y_1, ..., y_n)$ ein Pfad ohne Zurückverfolgung. Da g Nachbarknoten erhält, werden die y_i auf eine Folge von aneinanderhängenden Kanten abgebildet, d.h. $t(gy_i) = o(gy_{i+1})$. Da g bijektiv auf ungerichteten Kanten wirkt, werden keine Kanten von $(gy_1, ..., gy_n)$ direkt zurückgelaufen. Somit ist $(gy_1, ..., gy_n)$ ein Pfad ohne Zurückverfolgung von Länge n. □

Die nächsten Lemmata sind so formuliert, dass wir später Satz 2.2.10 und 2.2.12 gleichzeitig zeigen können.

Lemma 2.2.4. *Sei $T = (X, Y)$ ein Graph, auf dem eine Gruppe G ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.*
Sei $p \in X$ ein beliebiger Knoten und $neigh(p) := \{x \in X : \exists y \in Y, o(y) = p, t(y) = x\} = \{p_i : i \in I\} \subseteq X$ seine Nachbarknoten. Bezeichne mit G_p den Stabilisator von p unter der Operation G. Wähle für jeden Nachbarknoten p_i ein Gruppenelement $g_i \in G$, sodass $g_i p = p_i$.
Dann ist $G_p \cup \{g_i : i \in I\}$ ein Erzeugendensystem von G genau dann, wenn T zusammenhängend ist

Beweis. " $\Longleftarrow$ " : Sei $g \in G$ ein Element, das auf T operiert. g bildet den Knoten $p \in X$ auf $q := gp$ ab. Da T zusammenhängend ist, gibt es einen kürzesten Pfad $(y_1, .., y_n)$ von p zu q, welcher durch die Knoten $(p = p_0, p_1, ..., p_{n-1}, p_n = q)$ verläuft. Zeige nun durch Induktion über der Pfadlänge, dass es ein Gruppenelement $g_v = g_{i_1}...g_{i_n}$ gibt, dass als Produkt der Erzeuger geschrieben werden kann und p (entlang des Pfades) auf q verschiebt.
Falls $n = 0$, dann ist $p = q$ und es gilt $ep = q$.
Falls $n = 1$, dann ist q ein Nachbar von p. Es existert nach Definition des Erzeugendensystems ein g_i mit $g_i p = q$.
Falls $n > 1$: Nach Induktionsvoraussetzung existiert ein Produkt aus Erzeugern $g' = g_{i_1}...g_{i_{n-1}}$, sodass $g'p = p_{n-1}$. Es gilt nun, ein Gruppenelement zu finden, das p_{n-1} auf p_n abbildet. Man kann das Element p_{n-1} entlang des Pfades $(y_1, ..., y_{n-1})$ zurückziehen: $(g')^{-1}p_{n-1} = p$. Da Gruppenwirkungen Nachbarschaft erhalten, gilt $(g')^{-1}p_n \in neigh(p)$. Somit existiert ein Erzeugerelement g_{i_n} mit $g_{i_n}(g')^{-1}p_{n-1} = (g')^{-1}p_n \iff g_{i_n}p = (g')^{-1}p_n \iff (g'g_n)p = g_v p = p$, was ein Produkt der Erzeuger ist.
Wir sind noch nicht fertig. Auch wenn das Bild von p unter g und g_v gleich ist, können die beiden Gruppenelemente immer noch unterschiedlich auf anderen Knoten und Kanten wirken. Um das zu korrigieren, konstruieren wir nun eine "rotierende" Komponente g_r von g und zeigen, dass diese in G_p liegt (rotierend, nur im Sinne, dass es p festhält): $g_r := g_v^{-1}g$ mit $g_r p = g_v^{-1}gp = g_v^{-1}q = p \implies g_r \in G_p \implies g = g_v g_r$ ist ein Produkt von Erzeugern.

" $\Longrightarrow$ " : Der Quotient ist eine Schleife. Für ein $q \in X$ gibt es daher ein $g \in G$, sodass $q = gp$. g kann als Produkt $h_1...h_n$ der Erzeuger $G_p \cup \{g_i : i \in I\}$ geschrieben werden. Zeige die Hinrichtung des Lemmas durch Induktion über n.

Für $n = 0$ ist $q = ep$ also sind q und p verbunden.
Für $n = 1$ ist $q = g_r p$ für $g_r \in G_p$ oder $q = g_v p$ für $g_v \in \{g_i : i \in I\}$. In dem Fall ist q entweder p selbst oder ein Nachbarknoten
Für $n > 1$ gibt es nach Induktionsvoraussetzung einen Pfad von p zu $h_1 ... h_{n-1} p$. Da die Erzeugerelemente p entweder fix lassen oder auf einen Nachbar verschieben und da Gruppenwirkungen Nachbarschaft erhalten, sind $(h_1 ... h_{n-1})p$ und $(h_1 ... h_{n-1})h_n p$ durch eine Kante verbunden. Damit gibt es auch einen Pfad von p zu $p_n = q$. □

Lemma 2.2.5. *Sei $T = (X, Y)$ ein Graph, auf dem eine Gruppe G ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.*
Sei $p \in X$ ein beliebiger Knoten, y_+ eine beliebige Kante mit $o(y_+) = p$ und $G_p p, G_p y_+$ deren Bilder unter der Projektion in den Quotientengraphen T/G_p.
Dann hat $G_p p$ genau zwei angrenzende Knoten $G_p q_-, G_p q_+$
Es gibt zudem ein Element $s \in G$, das p entlang der Kante y_+ verschiebt und es gilt: $G_p q_- = G_p s^{-1} p$ und $G_p q_+ = G_p sp$.

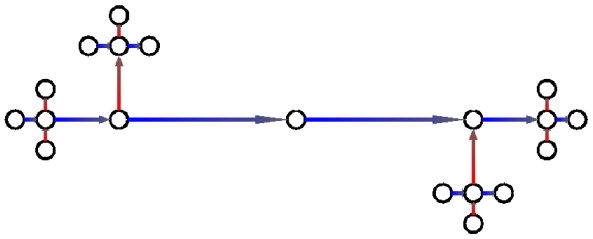


Abbildung 2.5: Der Quotient $V_4 *_{\langle h \rangle} /G_p$

Beweis. Seien $neigh(p) = \{x \in X : \exists z \in Y, o(z) = p, t(z) = x\} \subseteq Y$ alle Knoten, die an p angrenzen und $neigh(Gp) = \{Gx \in X/G : \exists Gz \in Y/G, o(Gz) = Gp, t(Gz) = Gx\} \subseteq Y$ alle Knoten, die an Gp angrenzen
Zeige, dass $G_p p$ mindestens zwei angrenzende Knoten hat: Sei $q \in neigh(p)$ durch y mit p verbunden. Es gibt ein $s \in G$, das p nach q verschiebt, d.h. $sp = q$. Für den Knoten $s^{-1}p$ gilt dann $s^{-1}p \notin G_p q$, da es sonst ein $g_r \in G_p$ gäbe mit: $g_r q = s^{-1}p \implies g_r^{-1} s^{-1} p = q$ und $g_r^{-1} s^{-1} q = g_r^{-1} p = p$. $g_r^{-1} s^{-1}$ vertauscht also die die Nachbarsknoten p, q und ist damit eine Operation mit Inversion.
Zeige, dass $G_p p$ maximal zwei angrenzende Knoten hat: Nach dem vorherigen Teil hat $G_p p$ mindestens zwei Nachbarn, nämlich $G_p(sp)$ und $G_p(s^{-1}p)$. Angenommen, es gäbe einen dritten Nachbar $G_p \rho$ für ein $\rho \in neigh(p)$. Der Knoten ρ ist durch eine Kante $y_\rho, o(y_\rho) = p, t(y_\rho) = \rho$ mit p verbunden. Da G transitiv auf allen Kanten wirkt[1], gibt es ein $g_\rho \in G$, sodass $g_\rho y_\rho = y_+$ oder $g_\rho y_\rho = y_- :=$, wobei $y_- := s^{-1}\bar{y}$ aus y_+ durch Inversion

[1]Da eine Schleife streng genommen aus zwei gerichteten Kanten y und $\bar{y}$ besteht, wirkt G nicht transitiv auf gerichteten Kanten. Solange wir jedoch berücksichtigen, dass jede gerichtete Kante auf y oder $\bar{y}$ abgebildet werden kann, bleibt das Argument korrekt

und anschließende Verschiebung mit s^{-1} hervorgeht.
Falls $g_\rho y_\rho = y_+$, dann ist $g_\rho \rho = t(g_\rho y_\rho) = t(y_+) = sp \implies Gr = G(sp)$.
Falls $g_\rho y_\rho = y_-$, dann ist $g_\rho \rho = t(g_\rho y_\rho) = t(y_-) = t(s^{-1}\bar{y}) = s^{-1}o(y) = s^{-1}p \implies G\rho = G(s^{-1}p)$.
Es kann also keinen dritten Nachbarn von $G_p p$ geben. □

Im Beweis des letzten Lemmas beobachten wir etwas interessantes. Die Bedingung, dass der Graph ohne Inversion operiert, war notwendig, um zu garantieren, dass $G_p p$ mindestens 2 Nachbarn hat. Die Bedingung der Schleife war notwendig, um zu zeigen, dass $G_p p$ maximal 2 Nachbarn hat.
Knoten aus $G_p sp$, $G_p s^{-1} p$ werden durch das s^{-1} und das s auf $G_p p$ verschoben. Das s steht dabei, wie wie in Kürze sehen werden, für ein stabiles Zeichen einer HNN-Erweiterung. Man kann zeigen, dass Gruppenwirkungen auf Bäumen, deren Quotienden Mehrfachschleifen (wie in Abb. 1.5) sind, mehr Nachbarn als 2 Nachbarknoten von $G_p p$ erlauben und mit mehreren stabilen Zeichen, d.h. mit mehrfachen HNN-Erweiterungen assoziiert werden können.
Im letzten Lemma führten wir Bezeichnungen für spezielle Objekte ein. Da wir auf diese Bezeichnungen nachfolgend häufiger zurückgreifen werden, ist es sinnvoll, diese eigenständig zu definieren.

Definition 2.2.6. Sei $T = (X, Y)$ ein Graph, auf dem eine Gruppe G ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.
Ab sofort steht p für einen willkürlichen aber festen Knoten aus X. $y_+ = y_{+1}$ ist eine beliebige aber feste Kante mit $o(y_+) = p$.
Bezeichne mit $s \in G$ das in Lemma 2.2.5 genannte Element, welches p entlang y_+ verschiebt. $y_- = y_{-1} := \overline{s^{-1}y}$ bezeichnet die y_+ *gegenüberliegende Kante.*
Um zukünftig Fallunterscheidungen zu vermeiden, schreiben wir für y_+ und y_- auch y_ε mit $\varepsilon = \pm 1$.

Korollar 2.2.7. *Sei $T = (X, Y)$ ein Graph, auf dem eine Gruppe G ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife. Seien p, s wie in Definition 2.2.6.*
Es gilt $s^{\varepsilon_1} g y_{\varepsilon_2} = \bar{y}_{\varepsilon_1}$ mit $\varepsilon_i = \pm 1, g \in G_p$ genau dann, wenn $\varepsilon_1 \neq \varepsilon_2$ und $g \in G_{y_{\varepsilon_2}}$

Beweis. " $\implies$ ": Angenommen $\varepsilon_1 = \varepsilon_2$, d.h. $s^{\varepsilon_1} g y_{\varepsilon_1} = \bar{y}_{\varepsilon_1}$, dann wirkt $s^{\varepsilon_1} g$ mit Inversion auf T. Daher gilt $\varepsilon_1 \neq \varepsilon_2$.
$s^{\varepsilon_1} g y_{\varepsilon_2} = \bar{y}_{\varepsilon_1} = s^{\varepsilon_1} y_{\varepsilon_2} \iff g y_{\varepsilon_2} = y_{\varepsilon_2} \iff g \in G_{y_{\varepsilon_2}}$.
" $\impliedby$ ": Gelte $\varepsilon_1 \neq \varepsilon_2, g \in G_{y_{\varepsilon_2}}$. Dann ist $s^{\varepsilon_1} g y_{\varepsilon_2} = s^{\varepsilon_1} y_{\varepsilon_2} = \bar{y}_{\varepsilon_1}$. □

Lemma 2.2.8. *Sei $T = (X, Y)$ ein Graph, auf dem eine Gruppe G ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.*
Dann ist $G_p \cup \{s\}$ ein Erzeugendensystem von G genau dann, wenn T zusammenhängend ist.

Beweis. Sei p, s wie in Definition 2.2.6. Sei $G_p \cup \{g_i : i \in I\}$ das Erzeugendensystem wie in Lemma 2.2.4 definiert.
Schreibe einen Nachbarsknoten $p_i \in neigh(p)$ als $g_i p$. Da $p_i \in G_p s^{-1} p$ oder $p_i \in G_p sp$ gilt $g_i = g_r s$ oder $g_i = g_r s^{-1}$ für ein $g_r \in G_p$. Somit ist $G_p \cup \{s, s^{-1}\}$ ein Erzeugendensystem von G.
Die Erzeugendensysteme $G_p \cup \{g_i : i \in I\}$ und $G_p \cup \{s, s^{-1}\}$ erzeugen die gleiche Gruppe. Damit gilt die Äquivalenz nach Lemma 2.2.4. □

Wir konnten von wenigen Informationen darüber, wie eine Gruppe G auf einem Graphen wirkt, ein kleines Erzeugendensystem angeben, das die Gruppe erzeugt.
Der nächste Schritt ist, eine Normalform aus den Erzeugern zu finden. Normalformen sind üblicherweise hilfreich, um zu identifizieren, wann Wörter der Erzeuger als das gleiche Gruppenelement evaluieren. In unserem Fall wird die Normalform strukturgleich mit Brittons Normalform für HNN-Erweiterungen sein. Daraus schließen wir, dass unsere Gruppe eine HNN-Erweiterung ist.
Um die Verbindung zwischen Gruppenwirkung und Normalform aufzubauen, identifizieren wir einen Ausdruck einer Normalform mit einem Pfad ohne Zurückverfolgung durch den Baum, auf dem die Gruppe wirkt. Da Bäume zyklenfrei und zusammenhängend sind, ist die Normalform wohldefiniert. Wir illustrieren das generelle Vorgeben am Beispiel $V_4 *_{\langle h \rangle}$ (Abbildung: 2.6):
Wir suchen die Normalform eines Gruppenelements, das den mittleren Knoten auf den grünen Knoten verschiebt. Diese eindeutige Normalform erhalten wir, indem wir die Gruppenelemente, die den mittleren Knoten entlang des Pfades verschieben, geschickt zusammensetzen. Zuletzt multiplizieren wir den Ausdruck mit einem korrigierenden, "rotierenden" g_{n+1}.

Satz 2.2.9. *Sei $T = (X, Y)$ ein Graph, auf dem eine Gruppe H ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.*
Die Darstellung eines Elements wie in Brittons Lemma existiert und ist eindeutig genau dann, wenn T zusammenhängend und zykelfrei ist. Die Darstellung ist:

Sei p, y_+, y_- und s wie in Definition 2.2.6. Notiere $G = G_p, A = G_{y_+}, B = G_{y_-}$. Für gegebene Repräsentantensysteme R_A von G/A und R_B von G/B (mit dem neutralen Element als Repräsentant von A und B) kann jedes Element $h \in H$ eindeutig als Produkt geschrieben werden:

$$h = r_1 s^{\varepsilon_1} r_2 s^{\varepsilon_2} ... r_n s^{\varepsilon_n} g_{n+1}$$

mit $\varepsilon_i = \pm 1, g_{n+1} \in G$. Zudem $r_i \in R_A$, wenn $\varepsilon_i = +1$ und $r_i \in R_B$, wenn $\varepsilon = -1$. Außerdem enthält der obere Ausdruck kein Teilwort $s^{-1}es$ oder ses^{-1}.

Beweis. Der Beweis nutzt Teile des Beweises von Brittons Lemma aus [SW79].
Behauptung 1: p und $p_n := hp$ sind durch einen Pfad ohne Zurückverfolgung der Länge n verbunden. Der Pfad verläuft durch die konsekutiv benachbarten Knoten $(p_i)_{i \in \{0,...,n\}} = (r_1 s^{\varepsilon_1} ... r_i s^{\varepsilon_i} p)_{i \in \{0,...,n\}}$. Da wir bei T von einem Mehrfachgraphen ausgehen, der mehrere

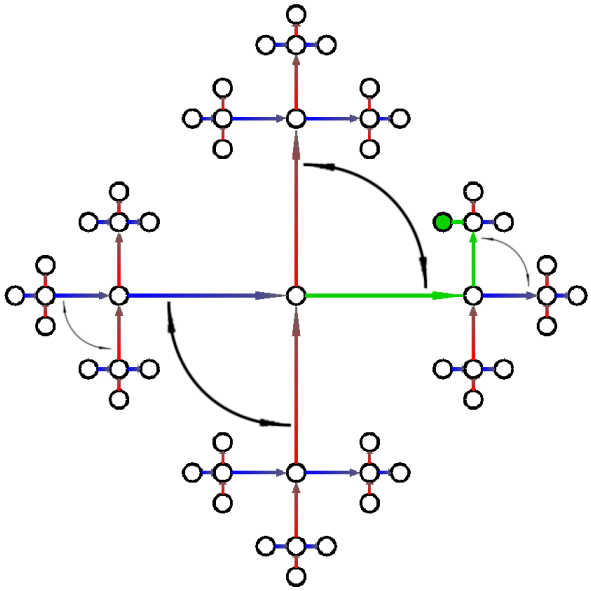


Abbildung 2.6: Äquivalenz von Zusammenhang+Zyklenfreiheit und Normalform bei $V_4 *_{\langle h \rangle}$

Kanten zwischen zwei Knoten haben kann, ist der Pfad durch die Knoten noch nicht eindeutig definiert. Wir definieren ihn daher durch:

$$(y_i)_{i\in\{1,\dots,n\}} = (r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i y_{\varepsilon_i})_{i\in\{1,\dots,n\}}$$

Wir zeigen, dass der Pfad zwischen den Knoten $(p_i)_{i\in\{0,\dots,n\}}$ verläuft.

$$\begin{aligned} o(r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i y_{\varepsilon_i}) &= r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i o(y_{\varepsilon_i}) \\ &= r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i p \\ &= r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} p \\ &= p_{i-1} \end{aligned}$$

$$\begin{aligned} t(r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i y_{\varepsilon_i}) &= r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i t(y_{\varepsilon_i}) \\ &= r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i (s^{\varepsilon_i} p) \\ &= p_i \end{aligned}$$

$(y_i)_{i\in\{1,\dots,n\}}$ ist ein zudem Pfad ohne Zurückverfolgung. Für eine Rückverfolgung muss nämlich gelten $y_{i+1} = \bar{y}_i$. d.h. $r_1 s^{\varepsilon_1} \dots r_i s^{\varepsilon_i} r_{i+1} y_{\varepsilon_{i+1}} = \overline{r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i y_{\varepsilon_i}} = r_1 s^{\varepsilon_1} \dots r_{i-1} s^{\varepsilon_{i-1}} r_i \bar{y}_{\varepsilon_i} \iff s^{\varepsilon_i} r_{i+1} y_{\varepsilon_{i+1}} = \bar{y}_{\varepsilon_i}$ Das ist nach Korollar 2.2.7 nur dann möglich, wenn $\varepsilon_{i+1} \neq \varepsilon_i$ und $r_{i+1} \in G_{y_{\varepsilon_{i+1}}} \implies r_{i+1} \in A$ oder $r_{i+1} \in B \implies r_{i+1} = e$, also wenn der Ausdruck ein Teilwort $s^{-1}es$ oder ses^{-1} enthält.

Behauptung 2: Es ist möglich, die Normalform von oben schrittweise zu konstruieren,

indem man wie in Behauptung 1 impliziert einem Pfad folgt. Sei $(y_1, ..., y_n)$ ein Pfad ohne Zurückverfolgung durch die Knoten $(p, p_1, ..., p_n)$ definiert. Dann gibt es bis auf ein variables $g_{n+1} \in G$ nur einen einzigen Ausdruck der Form

$$h = r_1 s^{\varepsilon_1} r_2 s^{\varepsilon_2} ... r_n s^{\varepsilon_n} g_{n+1}$$

sodass die Normalform-Voraussetzungen für r_i und ε_i, $hp = p_n$ und $r_1 s^{\varepsilon_1} ... r_{i-1} s^{\varepsilon_{i-1}} r_i y_{\varepsilon_i} = y_i$ für alle y_i des Pfades erfüllt sind
Zeige dies durch Induktion über die Pfadlänge n:
Für $n = 0$ ist $g_1 \in G$ das einzige Element der gesuchten Form.
Für $n = 1$ sind die Knoten p und p_1 Nachbarsknoten und es gibt ein eindeutiges $r_1 s^{\varepsilon_1}$ sodass $r_1 y_{\varepsilon_1} = y_1$ und $r_1 s^{\varepsilon_1} g_2 p = p_1$ mit $g_2 \in G$.
Für $n > 1$: Betrachte den Teilpfad $(y_1, ..., y_{n-1})$ zwischen p und p_{n-1}. Nach Induktionsvoraussetzung ist $h' = r_1 s^{\varepsilon_1} r_2 s^{\varepsilon_2} ... r_{n-1} s^{\varepsilon_{n-1}} g_n$ mit beliebigem $g_n \in G$ der einzige Ausdruck, der p entlang des Pfades verschiebt.
Wir versuchen nun p_{n-1} auf das benachbarte p_n abzubilden. Jedes Element, das dies tut, kann dargestellt werden als $h' r_n s^{\varepsilon_1} g_{n+1} h'^{-1}$ für ein bestimmtes $g_{n+1} \in G$. Durch Verkettung erhalten wir ein Gruppenelement, welches p erst auf p_{n-1} und dann auf p_n schiebt:

$$(h' r_n s^{\varepsilon_1} g_{n+1} h'^{-1}) h' = h' r_n s^{\varepsilon_1} g_{n+1} = (r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}}) g_n r_n s^{\varepsilon_n} g_{n+1} \tag{2.1}$$

Es gilt die Konjugationsrelation: Für ein $a \in A : \exists b \in B : s^{-1} a s = b \iff as = sb$ und analog $bs^{-1} = s^{-1} a$. Wir nutzen diese Relation, um 2.1 in die gewünschte Form zu bringen. Führe erstmal eine Fallunterscheidung durch:
Ist $\varepsilon_n = +1$, so schreiben wir $g_n r_n = r_a a$ für ein $r_a \in R_A, a \in A$. Es gibt ein $b \in B$, sodass gilt:

$$\begin{aligned} (r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}}) g_n r_n s^{\varepsilon_n} g_{n+1} &= (r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}}) r_a a s^{\varepsilon_n} g_{n+1} &(2.2)\\ &= r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}} r_a s^{\varepsilon_n} (b g_{n+1}) &(2.3)\end{aligned}$$

Im Fall $\varepsilon_n = -1$ kann man analog verfahren.
Der gerade hinzugefügte Term $r_a s^{\varepsilon_n} (b g_{n+1})$ soll aber nicht nur p_{n-1} auf p_n abbilden, sondern auch zusätzlich dem Pfad y_n folgen. D.h. er soll $y_n = (r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}}) r_a y_{\varepsilon_n}$ erfüllen. Bei fixem vorderem Teil $(r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}})$ existiert ein eindeutiges $r_A \in R_A$ sodass die Gleichung gilt.
Der Ausdruck 2.3 ist bis auf das hinterste $b g_{n+1}$ eindeutig. Würde man nämlich ein beliebiges Zeichen aus $r_1 s^{\varepsilon_1} ... r_{n-1} s^{\varepsilon_{n-1}} r_a s^{\varepsilon_n}$ verändern, so würde sich ebenfalls der aus Behauptung 1 resultierende Pfad verändern. Da der Ausdruck aus einem Pfad ohne Zurückverfolgung resultiert, gilt $y_{i+1} \neq \bar{y}_i \implies s^{\varepsilon_i} r_{i+1} y_{\varepsilon_{i+1}} \neq \bar{y}_{\varepsilon_i}$ und somit gibt es nach Korollar 2.2.7 kein Teilwort ses^{-1} oder $s^{-1}es$.

Wir zeigen nun die Äquivalenz von Existenz & Eindeutigkeit der Normalform und Zusammenhang & Zykelfreiheit des Graphen, auf dem die Gruppe operiert.
“ $\implies$ “ : Ist die Normalform für $h \in H$ existent und eindeutig, kann nach Behauptung 1

ein Pfad ohne Zurückverfolgung von p zu hp konstruiert werden. Es kann nur einen solchen Pfad geben, da man sonst nach Behauptung 2 eine zweite Normalform für h konstruieren könnte.
“ $\Longleftarrow$ “ : Ist der Baum zusammenhängend und zykelfrei, gibt es von einem Knoten p genau einen Pfad ohne Zurückverfolgung zu einem anderen Knoten hp für $h \in H$. Für diesen Pfad lässt sich nach Behauptung 2 eine eindeutige Normalform konstruieren. □

Satz 2.2.10. *Sei $T = (X, Y)$ ein Baum, auf dem eine Gruppe H ohne Inversion operiert. Sei zudem der Quotientengraph T/G eine Schleife.*
Dann ist H eine HNN-Erweiterung: $H \cong G_p *_{G_{y_+}}$ zu der Isomorphie $\varphi : G_{y_+} \to G_{y_-}, a \mapsto s^{-1}as$.

Beweis. Nach Satz 2.2.9 kann jedes Element $h \in H$ in Britton's Normalform geschrieben werden. Sei $G_p *_{G_y}$ eine HNN-Erweiterung. Nach Britton's Lemma kann jedes Element ebenfalls in einer eindeutigen Normalform geschrieben werden. Die beiden Normalformen sind strukturgleich, was nach einer Identifizierung der Erzeuger $\langle G_p \cup \{s\}\rangle$ eine bijektive Abbildung $\Phi : G_p *_{G_{y_+}} \to H$ induziert. Φ bildet ein Gruppenelement in Normalform aus $G_p *_{G_{y_+}}$ auf das Gruppenelement mit der gleichen Normalform aus H ab. Um zu zeigen, dass die Abbildung ein Isomorphismus ist, benötigen wir noch die Homomorphieeigenschaft.
Nach Identifikation der Erzeuger wie oben beobachten wir, dass Φ alle Relationen der HNN-Erweiterung $G_p *_{G_{y_+}}$ achtet: Für $g, h \in G_p$ gilt $\Phi(g)\Phi(h) = \Phi(gh)$ da $\Phi|_{G_p} = Id$. Für $a \in G_{y_+}$: $\Phi(s^{-1})\Phi(a)\Phi(s) = s^{-1}as = \varphi(a) = \Phi(\varphi(a))$, da $G_y, G_{s^{-1}y} \subseteq G_p$. Damit lässt sich $\Phi|_{G_p \cup \{s\}}$ auf den gesuchten Isomorphismus Φ fortsetzen. □

Wir wollen nun Satz 2.2.10 umkehren und zu einer HNN-Erweiterung einen Baum mit Gruppenoperation konstruieren.
Der letzte Satz und die vorherigen Lemmata zeigten, dass wenn eine Gruppe G auf einem Baum operiert, dass dann $G = G_p *_{G_y}$ als HNN-Erweiterung geschrieben werden kann mit $G \cong G_p$ Stabilisator eines Knotens $p \in X$ und $A \cong G_y$ Stabilisator einer Kante $y \in Y, o(y) = p$. In dem Falle gibt es wohldefinierte Bijektionen $H/G \to X, hG \mapsto hp$ und $H/A \to Y, hA \mapsto hy$.
Somit ist es naheliegend, einen Graphen auf folgende Weise zu definieren:

Definition 2.2.11. $\Gamma_H = (X, Y)$ sei der *Graph der HNN-Erweiterung $H = G*_A$* zu der Isomorphie $\varphi : A \to B$. Γ_H ist definiert durch:

$$X = H/G\,, Y = H/A \sqcup \overline{H/A}$$

Die Verbindungen zwischen Knoten und Kanten sind gegeben durch die Start- und Endfunktionen $o : Y \to X$, $t : Y \to X$. Es soll gelten $o(A) = G$ und $t(A) = sG$. Eine Verschiebung einer Kante durch ein Gruppenelement $h \in H$ soll Start- und Endknoten mit verschieben. Daher definieren wir

$$o(hA) = hG, t(hA) = hsG$$

und für die inversen Kanten:

$$o(\overline{hA}) = hsG, t(\overline{hA}) = hG$$

Die Start- und Endfunktionen sind wegen $A \subseteq G$ wohldefiniert.
Auf Γ_H gibt es eine natürliche Gruppenwirkung, definiert durch $g \cdot hG = (gh)G, g \cdot hA = (gh)A$ für $g, h \in H$. Die Gruppenwirkung operiert ohne Inversion, da eine Kante nicht durch Operationen auf ihr Inverses abgebildet werden kann. Der Quotientengraph ist eine Schleife, da die Gruppe transitiv auf H/A und transitiv auf $\overline{H/A}$ wirkt.

Wir konnten für die HNN-Erweiterung einen Graphen konstruieren. Damit dieser auch ein Baum ist, bleibt zu zeigen, dass er zusammenhängend und zykelfrei ist. Glücklicherweise waren die Lemmata allgemein genug formuliert, dass wir den Satz direkt formulieren können.

Satz 2.2.12. *Sei $H = G*_A$ eine HNN-Erweiterung. Dann gibt es einen Baum $T = (X, Y)$, auf dem $G*_A$ operiert.*

Beweis. Der Kandidat für den Baum ist der Graph der HNN-Erweiterung Γ_H.
Die Elemente von H können in Brittons Normalform geschrieben werden, damit ist Γ_H nach 2.2.9 zusammenhängend. Brittons Normalform von H ist eindeutig und damit ist Γ_H nach 2.2.9 zykelfrei. □

Kapitel 3

Konstruktion einer Gruppe mit unlösbarem Wortproblem

3.1 Turingmaschinen

Eine Turingmaschine ist ein mathematisches Modell eines Computers. Turingmaschinen können wie Computer programmiert werden, bestimmte Algorithmen durchzuführen und eignen sich damit hervorragend dazu, Algorithmen zu untersuchen oder herauszufinden, welche Aussagen überhaupt durch Algorithmen getroffen werden können.
Es gibt viele verschiedene Konventionen, eine Turingmaschine zu definieren. Für unsere Zwecke ist die Definition wie in [T$^+$36] am nützlichsten. Es wird davon ausgegangen, dass der Leser bereits mit Turingmaschinen vertraut ist. Im Folgenden wird die Turingmaschine nämlich nur eingeführt, um Parallelen mit der Modularen Maschine aufzuzeigen.

Definition 3.1.1. Wir definieren *Turingmaschinen* nach [T$^+$36] als eine Menge von *Konfigurationsübergängen* der Form (q_r, a_i, a_j, q_s, D) über einem endlichen Bandalphabet Σ und über einer endlichen Menge von Zuständen $\mathbb{Q}$ und fordern $q_r, q_s \in \mathbb{Q}, a_i, a_j \in \Sigma, D \in \{L, R\}$
Der Einfachheit halber identifizieren wir das Bandalphabet und die Zustandsmenge mit natürlichen Zahlen: $\Sigma = \{0, ..., n\}, \mathbb{Q} = \{n+1, ..., m\}$
Eine Turingmaschine soll zu jedem Zeitpunkt maximal einen möglichen Konfigurationsübergang besitzen. D.h. Für alle q_r, a_i gibt es maximal ein (q_r, a_i, a_j, q_s, D).

Definition 3.1.2. Eine *Konfiguration* $(...u_2u_1q\underset{\triangle}{a}v_1v_2...)$ umfasst die Information des Bandes $...u_2u_1av_1v_2...$, das Zeichen a der aktuellen Position $\underset{\triangle}{}$ auf dem Band und den aktuellen Automatenzustand der Maschine q.

Jeder Konfigurationsübergang gibt das Verhalten der Turingmaschine für eine bestimmte *Konfiguration* vor: Befindet sich der Automat der Turingmaschine im Zustand q_r und ließt das Zeichen a_i auf dem Band, so soll sie das Zeichen q_j auf das Band schreiben, in den Zustand q_s wechseln und das Band entweder nach links oder nach rechts verschieben.

Definition 3.1.3. Gibt es für eine Konfiguration keinen anwendbaren Übergang, nennen wir die Konfiguration *terminal*.

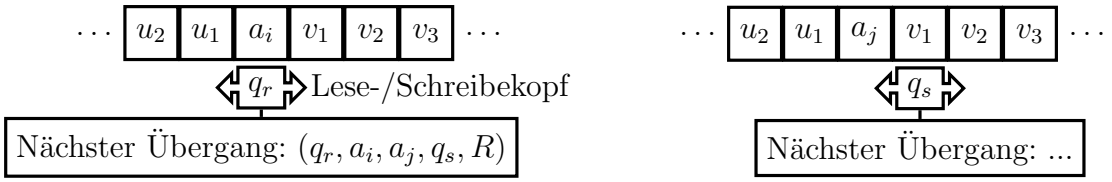


Abbildung 3.1: Anwendung eines Konfigurationsübergangs

Definition 3.1.4. Sei $\mathcal{T}$ eine Turingmaschine. Das *Halteproblem* ist die Menge der Eingabekonfigurationen

$$H_{\mathcal{T}} := \{(...u_2u_1qav_1v_2...) : \mathcal{T} \text{ hält für diese Eingabe}\}$$

Definition 3.1.5. Eine Aussage heißt *entscheidbar*, wenn es einen Algorithmus gibt, der beantworten kann, ob die Aussage gilt oder nicht.

Satz 3.1.6. *Es gibt eine Turingmaschine $\mathcal{T}$, sodass das Halteproblem $H_{\mathcal{T}}$ nicht entscheidbar ist.*

Beweis. Siehe [T$^+$36] □

Bemerkung 3.1.7. *Es existiert eine intuitive Fassung von Turings Beweis. Angenommen, das Halteproblem wäre entscheidbar. Dann existiert eine Turingmaschine H, die eine beliebige Turingmaschine T und eine beliebige Eingabe x "isst". H gibt 1 zurück, wenn T für die Eingabe x hält und 0, wenn T nicht hält.*
Wir konstruieren nun eine neue Maschine $H+$: $H+$ nimmt eine Turingmaschine T und eine Eingabe x entgegen und wirft sie in H ein. Gibt H 1 zurück, dann geht $H+$ in eine Endlosschleife, gibt H 0 zurück, dann hält $H+$.
Wir provozieren nun einen logischen Widerspruch, indem wir $H+$ als Turingmaschine und als Eingabe in sich selbst einsetzen. H berechnet nun, ob $H+$ bei seiner eigenen Eingabe hält oder nicht.
Angenommen $H+$ hält, dann gibt H den Wert 1 aus und $H+$ geht in eine Endlosschleife. Angenommen $H+$ hält nicht, dann gibt H den Wert 0 aus und $H+$ hält. Demnach kann das Halteproblem nicht entscheidbar sein!

3.2 Modulare Maschinen

Eine Modulare Maschine ist eine Maschine, die sich durch Abstraktion einer Turingmaschine ergibt. Das Bandalphabet wird wie eben durch die Zahlen $0, 1, ..., n$ und die Menge der Zustände durch die Zahlen $n + 1, ..., m - 1$ identifiziert. 0 steht hierbei für das *leere*

Feld des Bandes (Blank).
Da sowohl Turingmaschinen als auch Modulare Maschinen aus einer Menge von Konfigurationsübergängen bestehen, die zwischen Konfigurationen wechseln, unterscheiden sich die Maschinen nur geringfügig voneinander. Wir führen hier Modulare Maschinen wie in [AC80] ein.

Definition 3.2.1. Eine *Konfiguration einer Modularen Maschine* enthält Band und Zustandinformationen. Sie enthält die Informationen

- des Bandes links des Zeigers $u = \sum_{i=0}^{\infty} u_1 m^i \in \mathbb{N}$:
- des aktuellen Zeichens unter dem Zeiger $a \in \{0, ..., n\}$
- des aktuellen Zustands des Automaten $q \in \{n+1, ..., m-1\}$
- des Bandes rechts des Zeigers $v = \sum_{i=0}^{\infty} u_1 m^i \in \mathbb{N}$

Die Informationen u, a, q, v werden als ein 2-Tupel $(um + a, vm + q) \in \mathbb{N}^2$ oder $(um + q, vm + a) \in \mathbb{N}^2$ verkürzt. Beide Darstellungen einer Konfiguration sind austauschbar und wir werden beide je nach Kontext nutzen. Beobachte, dass aufgrund von $a < q$ die linke Notation für Konfiguration aus der rechten rekonstruiert werden kann und umgekehrt.

Die Bandinformationen u und v werden als Zahlen eines n-ären Stellenwertsystems kodiert. Hierbei werden die Bandfelder nah am Zeiger mit Stellen niedriger Potenz identifiziert. Die Kodierung mit einem Stellenwertsystem ist möglich, da zu jedem Zeitpunkt nur endlich viele Felder des Bandes mit nicht-leeren Zeichen beschrieben sind. Die Modulare Maschine wird in Abbildung 3.2 wie die Turingmaschine visualisiert.
Die Tatsache, dass wir zwei Darstellungen für die gleiche Konfiguration wählen können,

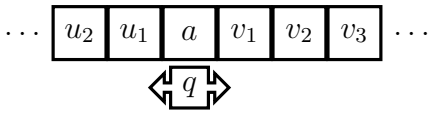


Abbildung 3.2: Konfiguration $(um + a, vm + q)$

weist auf einen wesentlichen Unterschied zwischen Turingmaschinen und Modularen Maschinen hin: Turingmaschinen unterscheiden zwischen Bandzeichen und Zuständen. Bei der Modularen Maschine hingegen sind Bandzeichen und Zustände austauschbar. Nach Vergessen von n, sind sie sogar ununterscheidbar.

Definition 3.2.2. Ein *Konfigurationsübergang einer Modularen Maschine* enthält die Informationen

- des gelesenen Bandzeichens $a \in \{0, ..., n\}$
- des zu schreibenden Bandzeichnes $a' \in \{0, ..., n\}$
- des aktuellen Automatenzustandes der Maschine $q \in \{n+1, ..., m-1\}$

- des neuen Automatenzustandes der Maschine $q' \in \{n+1, ..., m-1\}$
- der Bewegungsrichtung des Bandes $D \in \{L, R\}$

Die Informationen werden durch 2 austauschbare 4-Tupel $(q, a, c, D) = (q, a, a'm + q', D)$ bzw. $(a, q, c, D) = (a, q, a'm + q', D)$ verkürzt.

Definition 3.2.3. Eine *Modulare Maschine* besteht aus $m \in \mathbb{N}$ Zeichen sowie einer Menge von Konfigurationsübergängen der Form (a, b, c, D) mit $0 \leq a, b < m$, $0 \leq c < m^2$, $D \in \{L, R\}$. Jeder Konfigurationsübergang ist eindeutig, d.h. für jedes a, b gibt es maximal ein 4-Tupel, welches mit a, b beginnt.

Definition 3.2.4. Die *Anwendung eines Konfigurationsübergangs mit Rechtsbewegung* $(a, b, a'm + q', R)$ auf die Konfiguration $(\alpha, \beta) = (um + a, vm + b)$ ergibt die neue Konfiguration $(um^2 + a'm + q', v)$.

Die *Anwendung eines Konfigurationsübergangs mit Linksbewegung* (a, b, a', q', L) auf die Konfiguration $(\alpha, \beta) = (um + a, vm + b)$ ergibt die neue Konfiguration $(u, vm^2 + a'm + q')$.

Die Anwendung eines Konfigurationsübergangs von (α_1, β_1) zu (α_2, β_2) durch eine Modulare Maschine $\mathcal{M}$ wird mit $(\alpha_1, \beta_1) \overset{\mathcal{M}}{\to} (\alpha_2, \beta_2)$ bezeichnet.

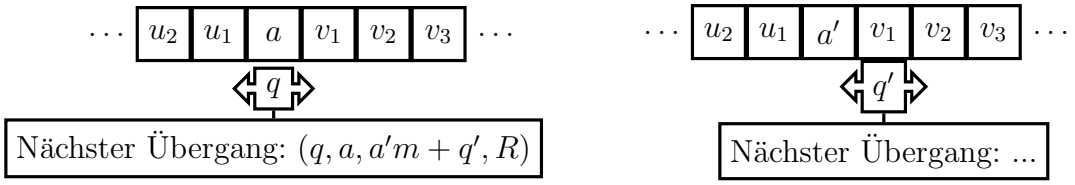


Abbildung 3.3: Anwendung eines Konfigurationsübergangs mit Rechtsbewegung

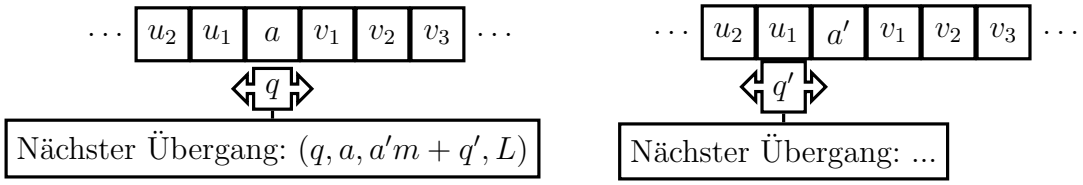


Abbildung 3.4: Anwendung eines Konfigurationsübergangs mit Linksbewegung

Definition 3.2.5. Eine Konfiguration, auf die kein Konfigurationsübergang angewendet werden kann, bezeichnen wir wie bei der Turingmaschine als *terminal*.

Definition 3.2.6. Nach jeder Anwendung eines Konfigurationsübergangs lässt sich so lange ein eindeutiger Konfigurationsübergang anwenden, bis eine terminale Konfiguration erreicht ist. Existiert eine endliche Folge von Anwendungen

$$(\alpha_1, \beta_1) \overset{\mathcal{M}}{\to} ... \overset{\mathcal{M}}{\to} (\alpha_n, \beta_n)$$

mit (α_n, β_n) terminal, so nennt man sie eine *Berechnung* von $\mathcal{M}$ und kürzt sie mit $(\alpha_1, \beta_1) \overset{\mathcal{M}^*}{\to} (\alpha_n, \beta_n)$ ab.
Wir bezeichnen (α_n, β_n) als die *Ausgabe von* $\mathcal{M}$ *zur Eingabe* (α_1, β_1).

Definition 3.2.7. Das *Halteproblem* einer Modularen Maschine $\mathcal{M}$ sind alle Eingabekonfigurationen, für die eine Berechnung existiert, d.h. die nach endlich vielen Konfigurationsübergängen in einer terminalen Konfiguration enden.

$$H_{\mathcal{M}} := \left\{(\alpha_1, \beta_1) : (\alpha_1, \beta_1) \overset{\mathcal{M}^*}{\to} (\alpha_n, \beta_n), (\alpha_n, \beta_n) \text{ terminal}\right\}$$

Satz 3.2.8. *Für jede Turingmaschine $\mathcal{T}$ kann eine Modulare Maschine $\mathcal{M}$ konstruiert werden, sodass die Ausgabekonfigurationen von $\mathcal{T}$ und $\mathcal{M}$ für eine Eingabekonfiguration gleich sind.*
Eine Konfiguration $(...u_2u_1qav_1v_2...)$ einer Turingmaschine ist gleich den beiden Konfigurationen einer Modularen Maschine $(u_{\mathcal{M}}m+a_{\mathcal{M}}, v_{\mathcal{M}}m+q_{\mathcal{M}})$ und $(u_{\mathcal{M}}m+q_{\mathcal{M}}, v_{\mathcal{M}}m+a_{\mathcal{M}})$, wenn

- *die aktuellen Zustände der Maschinen gleich sind: $q_{\mathcal{M}} = q$*
- *die Zeichen unter dem Lese-/Schreibkopf gleich sind: $a_{\mathcal{M}} = a$*
- *die Zeichen links des Lese-/Schreibkopfes gleich sind: $u_{\mathcal{M}} = \sum_{i=1} u_i m^i$*
- *die Zeichen rechts des Lese-/Schreibkopfes gleich sind: $v_{\mathcal{M}} = \sum_{i=1} v_i m^i$*

Beweis. Sei $\mathcal{T}$ definiert als Menge von Konfigurationsübergängen der Form (q_r, a_i, a_j, q_s, D) über einem Bandalphabet Σ und Zuständen $\mathbb{Q}$. Sei $\mathcal{M}$ die Modulare Maschine mit $m := |\Sigma| + |\mathbb{Q}|$ bestehend aus den Konfigurationsübergängen

$$\{(a_i, q_r, a_j m + q_s, D), (q_r, a_i, a_j m + q_s, D) : \text{Für } (q_r, a_i, a_j, q_s, D) \text{ aus } \mathcal{T}\}$$

Für eine gegebene Eingabekonfiguration führt die Anwendung des nächsten Konfigurationsübergang bei der Turingmaschine und bei der Modularen Maschine in eine gleiche, neue Konfiguration. Nach Induktion über die Zahl der Rechenschritte sind damit auch die Ausgabekonfigurationen gleich. □

Korollar 3.2.9. *Es gibt eine Modulare Maschine $\mathcal{M}$, sodass das Halteproblem $H_{\mathcal{M}}$ nicht entscheidbar ist.*

Beweis. Es existiert eine Turingmaschine, für die das Halteproblem nicht entscheidbar ist. Diese Turingmaschine lässt sich in eine Modulare Maschine $\mathcal{M}$ umwandeln. Wäre bei der resultierendenden Modularen Maschine das Halteproblem entscheidbar, so hätte man eine Möglichkeit gefunden, das Problem für die Turingmaschine entscheidbar zu machen. Demnach ist das Halteproblem für $\mathcal{M}$ nicht entscheidbar. □

Korollar 3.2.10. *Es gibt eine Modulare Maschine $\mathcal{M}$, sodass das Problem*

$$H_{\mathcal{M},0} := \left\{(\alpha_1, \beta_1) : (\alpha_1, \beta_1) \overset{\mathcal{M}^*}{\to} (0,0)\right\}$$

mit terminaler Konfiguration $(0,0)$ nicht entscheidbar ist.

Beweis. Sei $\mathcal{M}'$ eine Modulare Maschine, die aus einer Turingmaschine mit unlösbarem Wortproblem erzeugt wurde. Wir hängen an $\mathcal{M}'$ eine Modulare Maschine $\mathcal{N}'$ an, die das Band leert und dann in die Konfiguration $(0,0)$ wechselt, wenn $\mathcal{M}'$ hält. □

3.3 Kodierung der Maschine

Wit kommen nun zum Kern der Arbeit. Eine gegebene Modulare Maschine $\mathcal{M}$ soll nun wie im Paper von [Sim05] in eine Gruppe umgewandelt werden.
Wir definieren dazu eine spezielle Gruppe G. G enthält Untergruppen, die mit Konfigurationen von $\mathcal{M}$ korrespondieren. Durch HNN-Erweiterungen verbinden wir die Untergruppen miteinander. Dadurch simulieren wir die Konfigurationsübergänge von $\mathcal{M}$.

Definition 3.3.1. Unser Ausgang ist die *Gruppe der trivialen Modularen Maschine*, definiert durch:

$$G = \langle t, x, y \mid xy = yx \rangle$$

Die eben definierte Gruppe korrespondiert mit einer Modularen Maschine ohne Zustandsübergänge. Daher erhält sie ihren Namen.

Definition 3.3.2. Die *Untergruppe einer Konfiguration* $(\alpha, \beta) = (um + a, vm + q)$ ist die Untergruppe $T_{\alpha,\beta} := \langle t(\alpha, \beta) \rangle$ für $t(\alpha, \beta) = x^{-\alpha} y^{-\beta} t x^{\alpha} y^{\beta}$
Die *Untergruppe aller Konfigurationen* ist die Untergruppe $T = \langle t(\alpha, \beta) : \alpha, \beta \in \mathbb{Z} \rangle$

Um Berechnungen elegant zu halten, wird die Untergruppe einer Konfiguration nicht exakt als eine Untergruppe verstanden, die eine bestimmte Konfiguration kodiert. Während $(um+a, vm+q)$ und $(um+q, vm+a)$ (mit $a < q$) die gleiche Konfiguration bezeichnen, sind die Untergruppen $T_{um+a,vm+q}$ und $T_{um+q,vm+a}$ unterschiedlich zu behandeln. Wir nehmen diese Vereinfachung vor, damit wir für spätere Definitionen keine Fallunterscheidungen einbauen müssen.
Die Gruppe G und die Gruppe $T_{\alpha,\beta}$ lässt sich in einem Cayley-Graph visualisieren. (Abb. 3.5). G besitzt als Untergruppe $\langle x, y \rangle \cong \mathbb{Z}^2$, welches fett eingezeichnet ist. Von jedem Element aus $\mathbb{Z}^2$ zweigen zwei neue Kopien von $\mathbb{Z}^2$ ab. Die beiden Instanzen erreicht man

durch Anmultiplizieren von t bzw. t^{-1}.
Das Element $t(1,1) = x^{-1}y^{-1}tx^1y^1$ erhält man, indem man der dunklen roten Linie folgt. Die anderen roten Knoten bilden die Untergruppe $T_{1,1} = \langle t(1,1)\rangle$. Wie man erkennen kann, haben die Untergruppen $T_{\alpha,\beta}$ einer Konfiguration die Form eines "Turmes" über dem Element $x^\alpha y^\beta$.

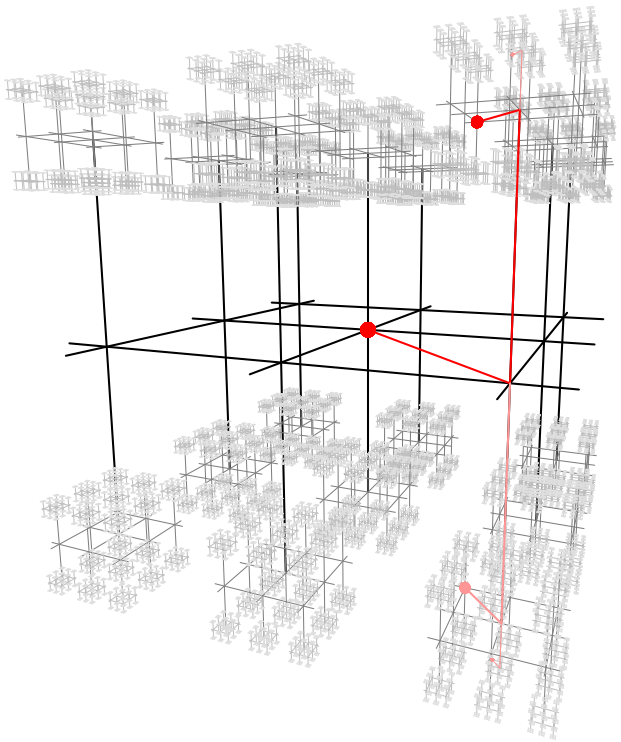


Abbildung 3.5: Der Cayley-Graph der Gruppe G

Um bei einer Turingmaschine oder einer Modularen Maschine den nächsten Konfigurationsübergang anzuwenden, sind die Informationen des Bandes links und rechts vom Lese-/Schreibkopf nicht notwendig. Das motiviert uns, eine Untergruppe zu definieren, welche Bandzeichen, die weit genug vom Lesekopf entfernt sind, vergisst.

Definition 3.3.3. Die *reduzierte Untergruppe einer Konfiguration* $(\alpha,\beta) = (um^l+u', vm^r+v') = (\sum_{i=l}^{\infty} u_i m^i + \sum_{i=0}^{l-1} u_i m^i, \sum_{i=r}^{\infty} v_i m^i + \sum_{i=0}^{r-1} v_i m^i)$ ist definiert durch:

$$T^{l,r}_{u',v'} = T^{l,r}_{\sum_{i=0}^{l-1} u_i m^i, \sum_{i=0}^{r-1} v_i m^i} := \left\langle \left\{ t\left(um^l + \sum_{i=0}^{l-1} u_i m^i, vm^r + \sum_{i=0}^{r-1} v_i m^i \right) : u,v \in \mathbb{Z} \right\} \right\rangle$$

Schreibe $(\alpha,\beta) = (\sum_{i=0}^{\infty} u_i m^i, \sum_{i=0}^{\infty} v_i m^i)$ als $...u_1u_0\underset{\triangle}{\ }v_0v_1...$ (Unterscheide aktuellen Zustand und aktuellen Zeichen u_0, v_0 durch $u_0 < v_0$, g.d.w. u_0 Zeichen). Ist $l,r = 1$, so vergisst $T^{1,1}_{u_0,v_0}$ alle Informationen, die weiter als 1 Feld links des Dreiecks und weiter als 1 Feld rechts des Dreiecks liegen. $T^{1,1}_{u_0,v_0}$ enthält nämlich alle Untergruppen einer Konfiguration,

für die die beiden Felder links und rechts neben dem Dreieck gleich sind. $T^{1,1}_{u_0,v_0}$ ist daher die Untergruppe aller Konfigurationen der Form $*u_0 \underset{\triangle}{} v_0*$

Allgemeiner vergisst $T^{l,r}_{u',v'}$ alle Informationen, die weiter als l Felder links des Dreiecks und weiter als r Felder rechts des Dreiecks liegen. Es korrespondiert mit allen Konfigurationen der Form $*u_{l-1}...u_1u_0 \underset{\triangle}{} v_0v_1...v_{r-1}*$

Die Definition von $T^{l,r}_{u',v'}$ besteht aus unendlich vielen Erzeugern. Wir wünschen uns eine Gruppe, die wie $T^{l,r}_{u',v'}$ die Konfigurationsinformationen ohne Band kodiert, jedoch aus endlich vielen Erzeugern besteht. Hierfür definieren wir die vereinfachte Untergruppe einer Konfiguration:

Definition 3.3.4. Die *vereinfachte Untergruppe einer Konfiguration* $(\alpha, \beta) = (um^l + u', vm^r + v')$ ist definiert durch:

$$G^{l,r}_{u',v'} = \left\langle t(u', v'), x^{m^l}, y^{m^r} \right\rangle$$

Das l, r im Exponenten von $G^{l,r}_{u',v'}$ erfüllt den gleichen Zweck wie bei $T^{l,r}_{u',v'}$. Die Untergruppe korrespondiert mit allen Konfigurationen der Form $*u_{l-1}...u_1u_0 \underset{\triangle}{} v_0v_1...v_{r-1}*$.

Lemma 3.3.5. *Es gilt* $T^{l,r}_{u',v'} = T \cap G^{l,r}_{u',v'}$.

Beweis. " $\subseteq$ " : Für eine Konfiguration $(um^l + u', vm^r + v')$ ist $t(um^l + u', vm^r + v') = x^{-um^l+u'}y^{-vm^r+v'}tx^{um^l+u'}y^{vm^r+v'} = x^{-um^l}y^{-vm^r}t(u', v')x^{um^l}y^{vm^r} \in G^{l,r}_{u',v'}$.
" $\supseteq$ " : Es gilt $x^kt(\alpha, \beta) = t(\alpha - k, \beta)x^k$ und $y^kt(\alpha, \beta) = t(\alpha, \beta - k)y^k$ für $k \in \mathbb{Z}$. Somit kann jedes Element von $G^{l,r}_{a,b}$ in die Form $gx^{k_1m^l}y^{k_2m^r}$ mit $g \in T^{l,r}_{u',v'}$ (für $k_1, k_2 \in \mathbb{Z}$) umgewandelt werden.
Da G keine Relation enthält, durch die man das Zeichen t entfernen könnte, enthält ein Wort w aus den Generatoren von T: $\left\{t(\alpha, \beta) = x^{-\alpha}y^{-\beta}tx^{\alpha}y^{\alpha} : \alpha, \beta \in \mathbb{Z}\right\}$ genau dann kein t, wenn w das neutrale Element ist. Somit gilt $x^{k_1m^l}y^{k_2m^r} \notin T$ für alle $k_1, k_2 \in \mathbb{Z} \implies gx^{k_1m^l}y^{k_2m^r} \in T \iff k_1, k_2 = 0$. □

Lemma 3.3.6. *Es gibt einen Isomorphismus* $G^{l,r}_{u',v'} \cong G$. *In Folge sind jeweils zwei vereinfachte Untergruppen einer Konfiguration zueinander isomorph.*
Des Weiteren gibt es einen Isomorphismus zwischen der Gruppe aller Konfigurationen und der reduzierten Untergruppe einer Konfiguration: $T^{l,r}_{u',v'} \cong T$.

Beweis. Intuitiv hat $G^{l,r}_{u',v'}$ die gleiche Struktur wie G. Es besteht aus einer $\mathbb{Z}^2$-Ebene, bei der von jedem Punkt weitere Ebenen nach oben und unten abspalten.
Betrachte den Homomorphismus $\varphi : G \to G^{l,r}_{u',v'}$, induziert durch die Bilder der Erzeuger $\varphi(x) = x^{m^l}$ $\varphi(y) = y^{m^r}$, $\varphi(t) = t(u', v')$. Der Homomorphismus ist wohldefiniert, da er alle Relationen von G erhält: $\varphi(x)\varphi(y) = x^{m^l}y^{m^r} = \varphi(y)\varphi(x)$.
Aus einem Element aus den Erzeugern $t(u', v'), x^{m^l}, y^{m^r}$ kann ein eindeutiges Urbild rekonstruiert werden. Somit ist φ ein Isomorphismus.

Die gleiche Abbildung, eingeschränkt auf die Untergruppe T, ergibt einen Homomorphismus, der den Erzeuger $t(u,v), u,v \in \mathbb{Z}$ von T auf den Erzeuger $x^{-ux^l}y^{-vy^r}t(u',v')x^{ux^l}y^{vy^r} = t(um^l+u', vm^r+v')$ von $T^{l,r}_{u',v'}$ bijektiv abbildet. Somit ist $\varphi|_T : T \to T^{l,r}_{u',v'}$ ein Isomorphismus. □

3.3.6 ermöglicht das Bilden einer HNN-Erweiterung, die verschiedene $G^{l,r}_{u',v'}$ konjugiert.

Definition 3.3.7. Für eine gegebene Modulare Maschine

$$\mathcal{M} = \{(q_i, a_i, a_i'm + q_i', R) : i \in I\} \cup \{(q_j, a_j, a_j'm + q_j', L) : j \in J\}$$

konstruieren wir aus G eine Gruppe $G*_{\mathcal{M}}$ durch mehrfache HNN-Erweiterung:
Für jeden Konfigurationsübergang, der den Lese-/Schreibkopf nach rechts bewegt ($i \in I$) konjugieren wir mithilfe eines stabilen Zeichens r_i die beiden Untergruppen $\varphi_i : G^{1,1}_{a_i,q_i} \cong G^{2,0}_{a_i'm+q_i',0}$. Konkret fügen wir der Gruppe G die Generatoren r_j und die Relationen

$$\begin{aligned} r_i^{-1}t(a_i,b_i)r_i &= t(a_i'm+q_i',0) \\ r_i^{-1}x^M r_i &= x^{M^2} \\ r_i^{-1}y^M r_i &= y \end{aligned}$$

hinzu.
Für jeden Konfigurationsübergang, der den Lese-/Schreibkopf nach links bewegt ($j \in J$) konjugieren wir mithilfe eines stabilen Zeichens l_j die beiden Untergruppen $\psi_j : G^{1,1}_{a_j,q_j} \cong G^{0,2}_{0,a_j'm+q_i'}$. Konkret fügen wir der Gruppe G die Generatoren l_j und die Relationen

$$\begin{aligned} l_j^{-1}t(a_j,b_j)l_j &= t(0,a_j'm+q_i') \\ l_j^{-1}x^M l_j &= x \\ l_j^{-1}y^M l_j &= y^{M^2} \end{aligned}$$

hinzu.

In der letzten Definition konjugierten wir die Gruppen $G^{1,1}_{a_i,q_i} \to G^{2,0}_{a_i'm+q_i',0}$. Die Gruppen korrespondieren mit einer Menge von Konfigurationen und durch die Konjugation erhält man eine natürliche Abbildung zwischen Konfigurationen der Form $*a_i \underset{\triangle}{} b_i* \mapsto *a_i'q_i' \underset{\triangle}{} *$.
Wir zeigen später, dass das Konjugieren des Elements $t(\alpha,\beta)$ tatsächlich mit der Anwendung eines Konfigurationsübergangs $(\alpha,\beta) \overset{\mathcal{M}}{\to} (\alpha_1,\beta_1)$ verträglich ist.

Korollar 3.3.8. *Die Gruppe $G_{\mathcal{M}}$ ist endlich präsentiert.*

3.4 Unlösbarkeit des Wortproblems

Im letzten Abschnitt konstruierten wir für eine Modulare Maschine $\mathcal{M}$ eine Gruppe $G_{\mathcal{M}}$. $\mathcal{M}$ bezeichne im folgenden Abschnitt eine Modulare Maschine mit unlösbarem Problem $H_{\mathcal{M},0}$. Wir zeigen nun, dass das Wortproblem einer Gruppe in $G_{\mathcal{M}}$ nicht entscheidbar ist.

Definition 3.4.1. Definiere

$$T_{H_{\mathcal{M},0}} := \langle t(\alpha,\beta) : (\alpha,\beta) \in H_{\mathcal{M},0} \rangle \subseteq G$$

als die Untergruppe aller Konfigurationen, deren Ausgabe $(0,0)$ ist

Definition 3.4.2. Die Pseudo-HNN-Erweiterung einer Untergruppe $U \in G$ (wie in 1.2.11) bezüglich der HNN-Erweiterung $G*_{\mathcal{M}}$ wird mit $U*_{\mathcal{M}}$ bezeichnet.

Lemma 3.4.3. *Die Untergruppe $T_{H_{\mathcal{M},0}}$ ist gut (wie in 1.2.12 definiert) bezüglich der HNN-Erweiterung $G*_{\mathcal{M}}$. Somit gilt nach Britton's Lemma $T_{H_{\mathcal{M},0}} \subseteq T_{H_{\mathcal{M},0}}*_{\mathcal{M}}$.*

Beweis. Wir benutzen Lemma 3.3.5:

$$\begin{aligned}
\varphi_i(T_{H_{\mathcal{M},0}} \cap G^{1,1}_{a_i,b_i}) &= \varphi_i(T_{H_{\mathcal{M},0}} \cap T \cap G^{1,1}_{a_i,b_i}) \\
&= \varphi_i(T_{H_{\mathcal{M},0}} \cap T^{1,1}_{a_i,b_i}) \\
&= \varphi_i(\langle t(um+a_i, vm+b_i) : (um+a_i, vm+b_i) \in H_{\mathcal{M},0} \rangle) \\
&= \varphi_i(\langle t(um^2+a_i'm+q_i', v) : (um^2+a_i'm+q_i', v) \in H_{\mathcal{M},0} \rangle) \\
&= T_{H_{\mathcal{M},0}} \cap T^{2,0}_{ma_i'+b_i,0} \\
&= T_{H_{\mathcal{M},0}} \cap T \cap G^{2,0}_{ma_i'+b_i,0} \\
&= T_{H_{\mathcal{M},0}} \cap G^{2,0}_{ma_i'+b_i,0} \\
&= T_{H_{\mathcal{M},0}} \cap \varphi(G^{1,1}_{a_i,b_i})
\end{aligned}$$

Der Beweis für die linksbewegenden ψ_j verläuft analog. □

Das folgende Lemma besagt, dass wir durch Rückwärtsanwendung von Konjugationen von $T_{0,0}$ alle Konfigurationen in $T_{H_{\mathcal{M},0}}$ aus der Konfiguration $(0,0)$ reproduzieren können.

Lemma 3.4.4. $T_{H_{\mathcal{M},0}}*_{\mathcal{M}} = T_{0,0}*_{\mathcal{M}}$

Beweis. $\supseteq$ ist einfach, da $T_{0,0} = \langle t(0,0) \rangle \subseteq T_{H_{\mathcal{M},0}} \subseteq T_{H_{\mathcal{M},0}}*_{\mathcal{M}}$.
Für $\subseteq$ ist es ausreichend, $t(\alpha,\beta) \in T_{0,0}*_{\mathcal{M}}$ für alle $(\alpha,\beta) \in H_{\mathcal{M},0}$ zu zeigen. $(\alpha,\beta) \in H_{\mathcal{M},0}$ gilt genau dann, wenn es eine Berechnung $(\alpha,\beta) \overset{\mathcal{M}}{\to} (\alpha_1,\beta_1) \overset{\mathcal{M}}{\to} ... \overset{\mathcal{M}}{\to} (0,0)$ gibt. Wir zeigen die Inklusion durch Induktion über die Länge der Berechnung. Für $(\alpha,\beta) = (0,0)$ gilt offensichtlich $t(\alpha,\beta) \in T_{0,0} \subseteq T_{0,0}*_{\mathcal{M}}$.
Angenommen $(\alpha,\beta) \overset{\mathcal{M}}{\to} (\alpha_1,\beta_1)$ ist eine Anwendung des Konfigurationsübergangs $(a_i, b_i, a_i'm + q_i', R)$. Es gilt

$$\begin{aligned}
t(\alpha,\beta) &= x^{-\alpha}y^{-\beta}tx^{\alpha}y^{\beta} \\
&= x^{-um-a_i}y^{-vm-b_i}tx^{um+a_i}y^{vm+b_i} \\
&= x^{-um}y^{-vm}\left(x^{-a_i}y^{-b_i}tx^{a_i}y^{b_i}\right)x^{um}y^{vm} \\
&= x^{-um}y^{-vm}t(a_i,b_i)x^{um}y^{vm}
\end{aligned}$$

und damit

$$\begin{aligned} r_i^{-1}t(\alpha,\beta)r_i &= r_i^{-1}\left(x^{-um}y^{-vm}t(a_i,b_i)x^{um}y^{vm}\right)r_i \\ &= \left(r_i^{-1}x^{-um}r_i\right)\left(r_i^{-1}y^{-vm}r_i\right)\left(r_i^{-1}t(a_i,b_i)r_i\right)\left(r_i^{-1}x^{um}r_i\right)\left(r_i^{-1}y^{vm}r_i\right) \\ &= x^{-um^2}y^{-v}t(c_i,0)x^{um^2}y^{v} \\ &= x^{-um^2-(a_i'm+q_i')}y^{-v}tx^{um^2+(a_i'm+q_i')}y^{v} \\ &= t(um^2+a_i'm+q_i',v) \\ &= t(\alpha_1,\beta_1) \end{aligned}$$

$t(\alpha_1,\beta_1) \in T_{0,0}*_{\mathcal{M}}$ nach Induktionsvoraussetzung $\implies t(\alpha,\beta) \in T_{0,0}*_{\mathcal{M}}$
Der Beweis für linksbewegende Konfigurationsübergänge verläuft analog. □

Die soeben gezeigte Gleichung $T_{H_{\mathcal{M},0}}*_{\mathcal{M}} = T_{0,0}*_{\mathcal{M}}$ impliziert, dass es nicht möglich ist, zu berechnen, ob ein Element $g \in G*_{\mathcal{M}}$ in $T_{0,0}*_{\mathcal{M}}$ liegt, da dies Aufschluss über das Halteproblem der Turingmaschine erlauben würde. Wir können uns diese Eigenschaft zu Nutze machen, um eine Gruppe mit unlösbarem Wortproblem zu definieren.

Satz 3.4.5. *Es gibt eine endlich präsentierte Gruppe, deren Wortproblem nicht entscheidbar ist.*

Beweis. Betrachte $G*_{\mathcal{M}}$. Bilde nun noch eine HNN-Erweiterung mit dem stabilen Zeichen k:

$$(G*_{\mathcal{M}})' := \left\langle G*_{\mathcal{M}}, k \mid k^{-1}hk = h, h \in T_{H_{\mathcal{M},0}}*_{\mathcal{M}} \right\rangle$$

Da $T_{0,0}*_{\mathcal{M}} = T_{H_{\mathcal{M},0}}*_{\mathcal{M}}$ endlich erzeugbar ist, ist $(G*_{\mathcal{M}})'$ endlich präsentierbar. Nach Brittons Lemma ist $k^{-1}gk = g \iff g \in T_{0,0}*_{\mathcal{M}}$. Insbesondere gilt $k^{-1}t(\alpha,\beta)k = t(\alpha,\beta) \iff t(\alpha,\beta) \in T_{0,0}*_{\mathcal{M}} = T_{H_{\mathcal{M},0}}*_{\mathcal{M}} \iff t(\alpha,\beta) \in T_{H_{\mathcal{M},0}} \iff (\alpha,\beta) \in H_{\mathcal{M},0}$.
Das Problem $k^{-1}t(\alpha,\beta)k = t(\alpha,\beta) \iff k^{-1}t(\alpha,\beta)k(t(\alpha,\beta))^{-1} = e$ lässt sich also auf das (unentscheidbare) Halteproblem von Modularen Maschinen zurückführen und ist damit ebenfalls nicht entscheidbar. □

Kapitel 4

Ausblick

Während unserer Behandlung von HNN-Erweiterungen und Modularen Maschinen sind einige Fragen offen geblieben, auf die nicht eingegangen werden konnte.

So könnte man die mehrfache HNN-Erweiterung einer Modulare Maschine betrachten und sich die Frage stellen, wie derartige mehrfache HNN-Erweiterungen auf Bäumen operieren. Die Anwort darauf ist ganz einfach: Operiert eine mehrfache HNN-Erweiterung ohne Inversion auf einem Baum, so ist dessen Quotient ein Knoten, mit einer Schleife für jedes stabile Zeichen der HNN-Erweiterung. Er sieht aus wie die Doppelschleife aus Abbildung 1.5. Die konstruierte Gruppe $G_{\mathcal{M}}$ ist durch Erweiterung von endlich vielen stabilen Zeichen l_i, r_j entstanden und so erwarten wir einen Quotienten bestehend aus endlich vielen Loops. Die Frage wie allgemeinere Gruppen auf Bäumen operieren und welche Quotientengraphen dabei herauskommen, wird in späteren Kapiteln von [Ser02] behandelt.

Interessant wäre auch die Frage, wie sich die Unlösbarkeit des Wortproblems auf Graphen überträgt, auf denen die Gruppe wirkt. Gibt es andere Möglichkeiten Gruppenwirkungen mit unlösbaren Wortpoblemen zu charakterisieren? Mit dieser Arbeit wurden Werkzeuge erarbeitet, die bei der Untersuchung dieser Frage sicherlich von Nutzen sind.

Zuletzt eröffnet die konstruierte Gruppe mit unlösbarem Wortproblem die Möglichkeit, neue Gruppen mit anderen unlösbaren Problemen zu konstruieren. Damit kann man beispielsweise zeigen, dass das Isomorphieproblem, also die Frage, dass zwei Präsentationen eine isomorphe Gruppe erzeugen, unlösbar ist. Man kann ebenfalls zeigen, dass das Konjugationsproblem, also die Frage, ob zwei Untergruppen zueinander konjugiert sind, unlösbar ist.

Literaturverzeichnis

[AC80] Stål Aanderaa and Daniel E Cohen. Modular machines, the word problem for finitely presented groups and collins' theorem. In *Studies in Logic and the Foundations of Mathematics*, volume 95, pages 1–16. Elsevier, 1980.

[BGBGL08] M Bridson, T Gowers, J Barrow-Green, and I Leader. Geometric and combinatorial group theory. *The Princeton companion to mathematics, T. Gowers, J. Barrow-Green, and I. Leader, Eds*, page 10, 2008.

[Car21] Nathan Carter. *Visual group theory*, volume 32. American Mathematical Soc., 2021.

[Ser02] Jean-Pierre Serre. *Trees*. Springer Science & Business Media, 2002.

[Sim05] Stephen G Simpson. A slick proof of the unsolvability of the word problem for finitely presented groups. *Preprint*, 2005.

[SW79] Peter Scott and Terry Wall. Topological methods in group theory. In *Homological group theory (Proc. Sympos., Durham, 1977)*, volume 36, pages 137–203, 1979.

[T$^+$36] Alan Mathison Turing et al. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.