Institut für Formale Methoden der Informatik
Abteilung Theoretische Informatik
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3610

Das Wortproblem für Omega-Terme über Zweivariablenlogik

Jan Philipp Wächter

Studiengang: Informatik

Prüfer: Prof. V. Diekert

Betreuer: Dr. M. Kufleitner

begonnen am: 16. Januar 2014

beendet am: 18. Juli 2014

CR-Klassifikation: F.2.2, F.4.1, F.4.3

Inhaltsverzeichnis

0 Über diese Arbeit

Kaum ein Maschinenmodell ist von so großer Bedeutung wie das der endlichen Automaten. Es findet nicht nur Anwendung in der technischen, theoretischen und praktischen Informatik, sondern auch in vielen anderen Wissenschaften und Wissenschaftsbereichen. Von daher mag es kaum verwundern, dass nicht nur die mit den endlichen Automaten untrennbar verbundenen regulären Sprachen, sondern auch viele darin enthaltene Sprachklassen von großem akademischen und praktischen Interesse sind. Neben der automatentheoretischen Untersuchung der regulären Sprachen gibt es daher noch andere Herangehensweisen: beispielsweise durch Logiken oder durch algebraische Strukturen. Dass es sich bei endlichen Automaten um recht einfache Maschinen handelt, überträgt sich auch auf die algebraischen Strukturen. So werden reguläre Sprachen durch endliche Monoide erkannt, einer algebraischen Struktur, die nur wenigen Axiomen gerecht werden muss.

Ein Sprachklasse unterhalb der regulären Sprachen erhält man beispielsweise, indem man sich auf Sprachen beschränkt, die durch Sätze der Prädikatenlogik erster Stufe oder *first-order logic* **FO** definierbar sind. Durch Beschränkung der Anzahl der in einem solchen Satz erlaubten Variablen lässt sich diese Sprachklasse weiter spezialisieren. Es ist bekannt, dass die Verwendung von nur drei Variablen keine Einschränkung darstellt. Die Untersuchung jener Sprachen, die sich mit Prädikatenlogik erster Stufe unter Verwendung von nur zwei Variablen, also durch die Zweivariablenlogik $\mathbf{FO}^2[<]$ definieren lassen, ergibt sich daher auf natürliche Weise.

Auf algebraischer Seite korrespondiert diese Klasse von Sprachen mit der Menge endlicher Monoide, genauer der *Varietät* endlicher Monoide **DA**. Jene Varietät **DA** erhält man auch durch Vereinigung der abzählbar unendlich hohen, sogenannten Trotter-Weil-Hierarchie. Diese besteht aus Ecken, $\vee$-Ebenen und $\cap$-Ebenen und ist zentraler Gegenstand der Betrachtungen dieser Arbeit.

Diese Betrachtungen befassen sich mit dem Wortproblem für ω-Terme von Varietäten in der Trotter-Weil-Hierarchie. Bei ω-Termen oder, wie sie hier genannt werden, π-Termen handelt es sich um Terme von Variablen, die mit einer zusätzlichen ${}^{\pi}$-Potenz versehen werden können. Anschließend können für die Variablen Elemente eines Monoids eingesetzt werden, die ${}^{\pi}$-Potenzen erhalten dann einen Wert, der als *unendlich oft* interpretiert werden kann. Anschaulich lässt sich ein π-Term daher mit dem Verhalten eines endlichen Automaten, dessen Ausführung nicht durch eine bestimmte Anzahl an Schritten beschränkt ist, verknüpfen. Zwei π-Terme u und v lassen sich zu einer Gleichung $u = v$ verknüpfen. Es ist dann möglich zu fragen, ob ein Monoid M diese Gleichung erfüllt. Dies ist dann der Fall, wenn sich links und rechts stets das selbe Monoidelement ergibt, sobald man für die Variablen beliebige Elemente einsetzt. Diese Fragestellung lässt sich erweitern: Erfüllt jedes Monoid in einer Varietät die Gleichung? Dabei handelt es sich um die Frage des Wortproblems für π-Terme der Varietät.

Diese Arbeit wird zeigen, dass sich das Wortproblem für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie durch eine logarithmisch platzbeschränkte nichtdeterministische Turingmaschine entscheiden lässt, dass es also zur Komplexitätsklasse NL gehört. Obwohl es sich bei NL um eine nichtdeterministische Platzklasse handelt, ist dieses Ergebnis interessant, da sich die Probleme aus NL effizient parallel entscheiden lassen. Eine Eigenschaft, die nicht zuletzt seit dem Aufkommen von Mehrkernprozessoren, eine immer größere Rolle spielt. Gleichzeitig liefert ein NL-Algorithmus auch einen deterministischen Polynomialzeitalgorithmus.

Der NL-Algorithmus für die Entscheidung des Wortproblems für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie lässt sich leicht abwandeln, um die Zugehörigkeit des Wortproblems für π-Terme von **DA** zu NL zu zeigen. Im Sinne der Komplexitätsklassen verbessert dies ein Ergebnis von A. Moura [10], das einen deterministischen Polynomialzeitalgorithmus zur Entscheidung dieses Problems liefert und damit Erkenntnisse von Almeida und Zeitoun erweitert, die in [2, 1] zeigten, dass sich das Wortproblem für π-Terme der Varietät der endlichen $\mathcal{R}$-trivialen Halbgruppen in Linearzeit entscheiden lässt. Die Varietät der endlichen $\mathcal{R}$-trivialen Monoide tritt auch als Ecke in der Trotter-Weil-Hierarchie auf.

Das Vorgehen in dieser Arbeit unterscheidet sich von dem Vorgehen in den genannten anderen Arbeiten. Das zentrale Konzept ist dabei das Einsetzen linearer Ordnungstypen für die $^{\pi}$-Potenzen der π-Terme, ähnlich wie in [5]. Auf die dabei entstehenden, im Allgemeinen unendlichen, verallgemeinerten Wörter werden dann einige Instrumente, die bei der Untersuchung der Trotter-Weil-Hierarchie sonst in einer endlichen Form vorkommen, übertragen. Neben der formellen Einführung und Wiederholung einiger grundlegender Begrifflichkeiten bildet dies den Inhalt des ersten Kapitels.

Das zweite Kapitel vertieft zunächst die Ergebnisse des ersten und führt anschließend eine bestimmte Normalisierung ein, die es erlaubt vom Unendlichen wieder ins Endliche überzugehen. Dies erlaubt dann die Skizzierung eines Algorithmus zur Entscheidung der besprochenen Wortprobleme für π-Terme am Ende des Kapitels.

Das dritte Kapitel schließlich zeigt durch Angabe entsprechender Algorithmen die Zugehörigkeit der Probleme zu NL. Dabei steht – wenig überraschend – die Speicherung benötigter Informationen in logarithmischem Platz im Mittelpunkt.

Im letzten Kapitel findet sich dann ein Ausblick auf angrenzende und weiterführende Fragestellungen rund um die Trotter-Weil-Hierarchie und das Wortproblem für π-Terme und schließlich eine Zusammenfassung der Ergebnisse.

1 Einführung

Am Anfang dieses Kapitels sollen einige Grundlagen vorgestellt und Konventionen eingeführt werden. Danach werden π-Terme und damit verbundene Begriffe eingeführt und untersucht und schließlich folgt ein Abschnitt über sogenannte Ranker und die Trotter-Weil-Hierarchie.

1.1 Grundlagen

Viele – wenn auch nicht unbedingt alle – der folgenden Begriffe sind dem Leser wahrscheinlich wohlbekannt. Dennoch lohnt es sich einige von ihnen genauer einzuführen, denn dies erlaubt es einige Vereinbarungen zu ihrem Gebrauch und zur Notation zu treffen. Außerdem finden sich in den folgenden Unterabschnitten auch einige Betrachtungen, die zunächst vertraut wirkende Begriffe aus einem anderen Blickwinkel präsentieren.

1.1.1 Funktionen und Notationen

Dieser Abschnitt wird die wichtigsten Konventionen zum Umgang mit Funktionen einführen, die in dieser Arbeit Verwendung finden.

Sei $f : A \to_p B$ eine partielle Funktion. Es ist also möglich, dass $f(a)$ für ein $a \in A$ nicht definiert ist. Da die Ausdrucksweise „ist nicht definiert“ recht umständlich ist, wird f erweitert zur (nicht partiellen) Funktion

$$\begin{aligned} f : A + \{\perp\} &\to B + \{\perp\} \\ \perp &\mapsto \perp \\ a \in A &\mapsto \begin{cases} f(a) & \text{falls } f(a) \text{ definiert ist} \\ \perp & \text{sonst,} \end{cases} \end{aligned}$$

wobei $A + B$ die disjunkte Vereinigung zweier Mengen A und B bezeichnet. Dies erlaubt es nun $f(a) = \perp$ statt „$f(a)$ ist nicht definiert“ zu schreiben. Diese Erweiterung wird im Folgenden stets implizit für alle auftretenden partiellen Funktionen verwendet. Dazu sollte man insbesondere beachten, dass für zwei partielle Funktionen $f : A \to_p B$ und $g : B \to_p C$ die Aussage $g(f(n)) = \perp$ bedeuten kann, dass f an der Stelle n nicht definiert ist oder dass $f(n)$ zwar definiert ist, g an der entsprechenden Stelle aber nicht.

In vielen Fällen ist es üblich – z. B. bei algebraischen Strukturen – für eine Funktion der Form $\cdot : A \times B \to C$ die Schreibweise $a \cdot b$ statt $\cdot(a, b)$ zu verwenden. Eine solche Funktion wird ab jetzt als *Verknüpfung* bezeichnet. Ist aus dem Zusammenhang zu erkennen, welche Verknüpfung $\cdot$ gemeint ist, wird auch ab statt $a \cdot b$ geschrieben, dies ist insbesondere für Verknüpfungen mit Bezeichnungen, die an Produkte erinnern, wie $\cdot$ oder $*$ der Fall. In

Unterabschnitt 1.1.5 wird eine ähnliche Notation für binäre Relationen vereinbart. Aus dem Kontext ist aber stets klar, ob es sich um eine Funktion oder eine binäre Relation handelt.

Auch andere im Umgang mit Funktionen übliche Schreibweisen finden in dieser Arbeit Anwendung. Ein Beispiel hierfür ist das folgende: Ist $f : A \to B$ eine Funktion und $A \subseteq A'$ eine Teilmenge von A, dann bezeichnet $f(A') = \{f(a) : a \in A'\}$ das Bild von A' unter f. Ebenso bezeichnet $f^{-1}(B') = \{a : f(a) \in B'\}$ die Menge der Urbilder einer Teilmenge $B' \subseteq B$ von B.

1.1.2 Halbgruppen und Varietäten

Betrachtet man die Menge ihrer Axiome, gehören Halbgruppen zu den einfachsten algebraischen Strukturen. Da sie bei vielen Problemen jedoch als natürliche Objekte auftreten, ist ihre genauere Untersuchung in vielen Bereichen von Interesse.

Bekanntermaßen handelt es sich bei einer Halbgruppe um eine Menge S mit einer assoziativen Verknüpfung $\cdot : S \to S$. Gibt es ferner ein neutrales Element in einer Halbgruppe, so heißt sie *Monoid*.

Konvention: Für eine Halbgruppe $(S, \cdot)$ wird im Folgenden oft einfach S geschrieben. Die Verknüpfung ergibt sich dann aus dem Kontext.

Andere wichtige Begriffe in diesem Zusammenhang werden als weitgehend bekannt vorausgesetzt. Darunter der der *Unterhalbgruppe* und des *Untermonoids*, insbesondere aber auch der des *Halbgruppenhomomorphismus* und des *Monoidhomomorphismus*.

Konvention: Wird aus dem Zusammenhang klar, ob ein Halbgruppen- oder ein Monoidhomomorphismus gemeint ist, so wird nur von einem *Homomorphismus* gesprochen.

Ein wichtiges Monoid ist das Monoid Σ^* der endlichen Wörter über einem Alphabet Σ. Ein *Alphabet* ist hierbei eine beliebige endliche Menge von *Buchstaben*. Eine Teilmenge $L \subseteq \Sigma^*$ wird – wie üblich – als *Sprache* endlicher Wörter über Σ bezeichnet. Eine solche Sprache L wird durch ein Monoid M *erkannt*, wenn es einen Homomorphismus $\varphi : \Sigma^* \to M$ mit $L = \varphi^{-1}(\varphi(L))$ gibt.

Aufgrund dieses Zusammenhangs zwischen Sprachen endlicher Wörter und Monoiden verwundert es nicht, dass sich bestimmte Abschlusseigenschaften von Mengen von Sprachen auf die zugehörige Menge der Monoide, die eine dieser Sprachen erkennen, übertragen. Dies führt auf den Begriff der *Varietät*, der vielleicht weniger bekannt ist als der Begriff des Monoids selbst, aber im Folgenden noch wichtig sein wird.

Definition (Varietäten endlicher Monoide)**:** Eine Menge $\boldsymbol{V}$ von endlichen Monoiden heißt *Varietät* endlicher Monoide[†], falls folgende Abschlusseigenschaften erfüllt sind:

- Ist $M \in \boldsymbol{V}$ und M' ein Untermonoid von M, so ist $M' \in \boldsymbol{V}$.
- Ist $M \in \boldsymbol{V}$ und $\varphi : M \to M'$ ein Monoidhomomorphismus, so ist $\varphi(M) \in \boldsymbol{V}$.
- Jedes endliche direkte Produkt von Monoiden aus $\boldsymbol{V}$ ist in $\boldsymbol{V}$.

Mehr zu Varietäten, z. B. wie sich diese Abschlusseigenschaften auf die Menge der von Monoiden in $\boldsymbol{V}$ erkannten Sprachen endlicher Wörter auswirken, findet sich in [12].

[†] oftmals auch *Pseudovarietät* genannt

Sind $\boldsymbol{V}$ und $\boldsymbol{W}$ zwei Varietäten, so bezeichnet $\boldsymbol{V} \vee \boldsymbol{W}$ die kleinste Varietät, die alle Monoide aus $\boldsymbol{V}$ und alle Monoide aus $\boldsymbol{W}$ enthält. Außerdem ist der Schnitt zweier Varietäten wieder eine Varietät, wie man leicht überlegt.

Ihrer Definition nach können Halbgruppen und Monoide sowohl endlich als auch unendlich groß sein. Im Folgenden werden die meisten von ihnen jedoch endlich sein, daher ist es sinnvoll diese Art von Halbgruppen genauer zu betrachten. Dabei ist vor allem das folgende Resultat von zentraler Bedeutung:

Lemma 1: Sei S eine endlich Halbgruppe. Für jedes Element $s \in S$ gibt es eine natürliche Zahl $m_s > 0$, sodass s^{m_s} idempotent ist, also dass $s^{2m_s} = s^{m_s}$ gilt. Außerdem gibt es eine natürliche Zahl $m > 0$, so dass s^m für alle Elemente $s \in S$ idempotent ist.

Beweis. Für den ersten Teil der Aussage siehe z. B. [4, S. 12]. Der zweite Teil ergibt sich durch Bilden des kleinsten gemeinsamen Vielfachen aller Potenzen m_s aus dem ersten Teil. □

Konvention: Für eine endliche Halbgruppe S bezeichnet $S!$ die kleineste natürliche Zahl > 0, sodass $s^{S!}$ für alle Elemente $s \in S$ idempotent ist. Es muss dabei nicht unbedingt $S! = |S|!$ gelten.

1.1.3 Graphen und Bäume

Obwohl viele Begriffe im Umfeld von Graphen und Bäumen eine weitgehend intuitive Bedeutung haben, gibt es dennoch oft Ausprägungen dieser Begriffe mit subtilen Unterschieden. Ziel diese Unterabschnitts ist es daher zu klären, was genau gemeint ist, wenn im weiteren Verlauf dieser Arbeit ein solcher Begriff auftaucht. Ein *Graph* G ist ein Paar (V, E) aus *Knotenmenge* V und *Kantenmenge* $E \subseteq V \times V$. Ist also von einem Graphen die Rede, so ist genaugenommen ein *gerichteter* Graph gemeint. Bei einem *beschrifteten* Graph wird das Paar zu einem Tripel (V, E, λ) ergänzt. Dabei ist $\lambda : V \to \Sigma$ die *Beschriftung* der Knoten des Graphen mit Buchstaben aus dem Alphabet Σ. Ist G ein Graph, so meint $v \in G$, dass v ein Knoten in G ist.

Zur Vereinfachung soll als Nächstes die folgende Konvention vereinbart werden:

Konvention: Ist aus dem Kontext klar, welcher Graph $G = (V, E)$ gemeint ist, wird im Folgenden für $(v, v') \in E$ einfach $v \to v'$ geschrieben.

Bei einem *Pfad* im Graphen G mit n Knoten handelt es sich nun um eine endliche Sequenz $p = (v_1, v_2, \ldots, v_n)$ von Knoten, sodass $v_i \to v_{i+1}$ für alle $1 \leq i < n$ gilt. Man beachte, dass auch ein einzelner Knoten als Pfad betrachtet werden kann! Der erste Knoten auf einem Pfad p wird als *Anfang*, der letzte als *Ende* und mögliche Knoten dazwischen als *innere Knoten* bezeichnet.

Ebenso wie ein Graph, lässt sich auch ein Pfad um eine Beschriftung ergänzen: Eine endliche Sequenz $p = ((v_1, a_1), (v_2, a_2), \ldots, (v_n, a_n))$ von Elementen aus $V \times \Sigma$ für ein beliebiges Alphabet Σ heißt *beschrifteter Pfad* in $G = (V, E)$ mit n Knoten. Ebenso wie bei beschrifteten Graphen bezieht sich die Beschriftung also auf die Knoten und nicht die Kanten. a_i heißt dabei *Beschriftung* von v_i (für $1 \leq i \leq n$). Anfang, Ende und innere

Knoten eines beschrifteten Pfades entsprechen denen des zugehörigen (unbeschrifteten) Pfades $(v_1, v_2, \ldots, v_n)$.

Konvention: Für einen Pfad $p = (v_1, v_2, \ldots, v_n)$ wird im Folgenden auch $p = v_1 \to v_2 \to \cdots \to v_n$ geschrieben und für einen beschrifteten Pfad $p = ((v_1, a_1), (v_2, a_2), \ldots, (v_n, a_n))$ auch $p = {}^{v_1}\!/_{a_1} \to {}^{v_2}\!/_{a_2} \to \cdots \to {}^{v_n}\!/_{a_n}$. Außerdem bezeichnet $(p)_i := v_i$ den i-ten Knoten von p für $1 \leq i \leq n$. Bei einem beschrifteten Pfad bezeichnet zudem $p(v_i) := a_i$ die Beschriftung a_i von v_i, sofern dies wohldefiniert ist.

Bei einem *Baum* $T = (V, E, v_0)$ mit *Wurzel* v_0 werden im Folgenden die Kanten stets von der Wurzel weg gerichtet, es handelt sich also um Out-Trees. Wie üblich wird ein Knoten mit Ausgangsgrad 0 als Blatt bezeichnet. Außerdem lassen sich auch die Knoten eines Baumes beschriften, das Ergebnis ist dann ein *beschrifteter Baum* $T = (V, E, v_0, \lambda)$, wobei $\lambda : V \to \Sigma$ wieder die Beschriftung der Knoten mit Buchstaben aus einem Alphabet Σ darstellt.

Mit dieser Betrachtungsweise gibt es in einem Baum für jeden Knoten v genau einen Pfad mit Anfang v_0 und Ende v. Daraus ergibt sich die Möglichkeit jeden Knoten im Baum mit dem (eindeutigen) Pfad von der Wurzel zu diesem Knoten zu identifizieren. Es gibt allerdings noch eine andere Möglichkeit der Betrachtung von Pfaden im Baum, dabei spielt folgende Aussage eine wichtige Rolle:

Lemma: Zwei Knoten in einem Baum sind über höchstens einen Pfad miteinander verbunden.

Beweis. Angenommen der Knoten u ist über zwei unterschiedliche Pfade p_1 und p_2 mit einem (möglicherweise anderen) Knoten v verbunden. Sei p_u der (eindeutige) Pfad, der an der Wurzel anfängt und in u endet. Das Verknüpfen von p_u mit p_1 und mit p_2 liefert zwei unterschiedliche Pfade, die an der Wurzel anfangen und an v enden. Dies ist ein Widerspruch. □

Dieses Lemma rechtfertigt also eine alternative Betrachtungsweise für Pfade in Bäumen. Ein Pfad $p = v_1 \to v_2 \to \cdots \to v_n$ im Baum $T = (V, E, v_0)$ lässt sich auch als partielle Funktion über den Knoten betrachten. Diese partielle Funktion ist genau auf den Knoten definiert, über die der Pfad verläuft. Da es keinen anderen Pfad mit v_1 als Anfang und v_n als Ende in G geben kann, beschreibt eine solche partielle Funktion also auch genau den Pfad p. Formal ausgedrückt wird p also mit der partiellen Funktion

$$\begin{aligned} p : V &\to_p \{\top\} \\ v &\mapsto \left\{ \begin{array}{ll} \top & \text{falls } v = v_i \text{ für ein } i \in \{1, 2, \ldots, n\} \end{array} \right. \end{aligned}$$

identifiziert. Auch ein beschrifteter Pfade $p = {}^{v_1}\!/_{a_1} \to {}^{v_2}\!/_{a_2} \to \cdots \to {}^{v_n}\!/_{a_n}$ über dem Alphabet Σ lässt sich so interpretieren. p wird dann als partielle Funktion

$$\begin{aligned} p : V &\to_p \Sigma \\ v &\mapsto \left\{ \begin{array}{ll} a_i & \text{falls } v = v_i \text{ für ein } i \in \{1, 2, \ldots, n\} \end{array} \right. \end{aligned}$$

aufgefasst. Man beachte, dass dies konsistent mit der oben eingeführten Schreibweise $p(v)$ für die Beschriftung eines Knotens v ist. Tatsächlich macht diese Betrachtung keinen

Unterschied zwischen einem Pfad und einem beschrifteten Pfad über dem einelementigen Alphabet $\{\top\}$. Es ist daher sinnvoll, diese Unterscheidung insgesamt fallen zu lassen.

1.1.4 Turing-Maschinen und Komplexitätsklassen

Der Begriff der Turing-Maschine ist in der Informatik von so zentraler Bedeutung, dass er an dieser Stelle wohl nicht wiederholt werden muss. Bekanntermaßen unterscheidet man zwischen deterministischen und nichtdeterministischen Turing-Maschinen. Erstere haben für jede Eingabe nur einen möglichen Berechnungspfad, der diese Eingabe entweder akzeptiert oder eben nicht akzeptiert, und letztere akzeptieren ihre Eingabe, wenn mindestens einer ihrer Berechnungspfade sie akzeptiert. Um auch Turing-Maschinen mit sublinearem Platzverbrauch untersuchen zu können, wird im Folgenden von einem Turing-Maschinen-Modell mit speziellem Eingabe-Band ausgegangen. Die Turing-Maschine besitzt für dieses Band lediglich einen in beide Richtungen bewegbaren Lesekopf, sie kann den Inhalt des Bandes aber nicht modifizieren.

Bei der Diskussion des Zeit- oder Platzbedarfs von Turing-Maschinen ist in der Regel nur das asymptotische Verhalten interessant. Wichtigstes Hilfsmittel dabei sind jene Symbole, für die Edmund Landau namensgebend war. Für diese Arbeit ist nur $\mathcal{O}$ über den nicht-negativen Zahlen von Bedeutung: Seien $f, g : \{0, 1, \dots\} \to \{0, 1, \dots\}$ zwei Funktionen. Dann liegt f in $\mathcal{O}(g)$, falls es eine reelle Konstante $c > 0$ und eine nicht-negative Zahl $n_0 \in \{0, 1, \dots\}$ gibt, sodass $f(n) \leq c \cdot g(n)$ für alle $n > n_0$ gilt.

Der Platzbedarf einer Turing-Maschine wird durch eine Funktion $s : \{0, 1, \dots\} \to \{0, 1, \dots\}$ beschränkt, wenn alle Berechnungspfade auf jeder Eingabe der Länge n höchstens $s(n)$ Arbeitsfelder belegen. Der Platzbedarf einer Turing-Maschine ist logarithmisch beschränkt, wenn ihr Platzbedarf durch eine Funktion s mit $s \in \mathcal{O}(\log n)$ beschränkt wird. Die Menge NL ist die Menge aller Entscheidungsprobleme, die durch eine im Platzbedarf logarithmisch beschränkte nichtdeterministische Turing-Maschine entschieden werden können. Als nichtdeterministische Platzklasse ist NL bekanntermaßen unter Komplement abgeschlossen [6, 15]. Um die Zugehörigkeit eines Entscheidungsproblems zu NL zu zeigen ist es also ausreichend einen nichtdeterministischen Algorithmus anzugeben, der zu einer logarithmisch platzbeschränkten nichtdeterministischen Turing-Maschine führt, bei der genau dann mindestens ein Berechnungspfad „nein“ ausgibt, wenn dies auch die Antwort auf das Entscheidungsproblem unter der selben Eingabe ist. Dieses Vorgehen findet in Kapitel 3 Anwendung.

1.1.5 Ordnungen und Ordnungstypen

Es gibt eine Vielzahl an verschiedenen Ausprägungen einer mathematischen Ordnung. Oft kommt es auch vor, dass für die selbe Ausprägung unterschiedliche Namen üblich sind. Dieser Unterabschnitt soll eine einheitliche Sprechweise für den Rest der Arbeit festlegen. Dabei treten viele bekannte, aber auch einige weniger bekannte Konzepte im Umfeld von Ordnungen auf. Am Anfang stehen zunächst einige einfache Definitionen.

Definition (Binäre Relation)**:** Sei P eine Menge. Eine Teilmenge $\mathcal{R} \subseteq P \times P$ heißt *binäre Relation* über P.

Konvention: Wie allgemein üblich wird im Folgenden bei einer binären Relation $\mathcal{R}$ für $(p, p') \in \mathcal{R}$ auch $p \mathrel{\mathcal{R}} p'$ und für $(p, p') \notin \mathcal{R}$ auch $p \not\mathrel{\mathcal{R}} p'$ geschrieben. Wie eingangs erwähnt, erinnert dies auch an die Schreibweise $a \cdot b$ für Verknüpfungen. Bei Unklarheiten ergibt sich jedoch aus dem Kontext, ob eine Relation oder eine Verknüpfung gemeint ist.

Definition (transitiv, reflexiv, antisymmetrisch, total)**:** Eine binäre Relation $\mathcal{R}$ über P heißt

- *transitiv*, falls $\forall p, q, r \in P : p \mathrel{\mathcal{R}} q$ und $q \mathrel{\mathcal{R}} r \implies p \mathrel{\mathcal{R}} r$ gilt,
- *reflexiv*, falls $\forall p \in P : p \mathrel{\mathcal{R}} p$ gilt,
- *irreflexiv*, falls $\forall p \in P : p \not\mathrel{\mathcal{R}} p$ gilt,
- *symmetrisch*, falls $\forall p, q \in P : p \mathrel{\mathcal{R}} q \implies q \mathrel{\mathcal{R}} p$ gilt,
- *antisymmetrisch*, falls $\forall p, q \in P : p \mathrel{\mathcal{R}} q$ und $q \mathrel{\mathcal{R}} p \implies p = q$ gilt und
- *total*, falls $\forall p, q \in P : p \mathrel{\mathcal{R}} q$ oder $q \mathrel{\mathcal{R}} p$ gilt.

Definition (partielle, lineare, strenge Ordnung, geordnete Menge)**:** Eine transitive, reflexive und antisymmetrische binäre Relation heißt *partielle Ordnung* oder Halbordnung. Eine totale partielle Ordnung heißt *lineare Ordnung* oder Totalordnung.
Eine transitive, irreflexive und antisymmetrische binäre Relation heißt *strenge Ordnung* oder Striktordnung.
Ist $\mathcal{R}$ eine partielle Ordnung über P, so heißt $(P, \mathcal{R})$ *partiell geordnete Menge*. Ist $\mathcal{R}$ eine lineare Ordnung über P, so heißt $(P, \mathcal{R})$ *linear geordnete Menge*.

Beispiel: $\leq$ ist eine lineare Ordnung über den natürlichen Zahlen.

Aus jeder partiellen Ordnung erhält man durch Entfernen der reflexiven Elemente eine strenge Ordnung (wie man sich leicht überlegt).

Konvention: Für partielle (und damit auch lineare) Ordnungen wird oft $\leq$ geschrieben. Dabei muss es sich nicht unbedingt um die natürliche $\leq$-Relation auf den reellen Zahlen oder einer ihrer Teilmengen handeln. Die zugehörige strenge Ordnung ist dann $<$. Im Zweifel ergibt sich aus dem Zusammenhang, welche Ordnung gemeint ist – die auf den reellen Zahlen oder eine andere.
Im üblichen Sinne wird im Folgenden auch $\geq$ und $>$ geschrieben.

Als nächste soll eine Konstruktion auf geordneten Mengen eingeführt werden, die sich im weiteren Verlauf als nützlich herausstellen wird. Eine partiell oder linear geordnete Menge wird dabei um ein kleinstes oder ein größtes Element ergänzt.

Definition (Ergänzung um ein kleinstes oder ein größtes Element)**:** Sei $(P, \leq)$ eine partiell oder linear geordnete Menge. Definiere[†]

$$\begin{array}{ll}
P_{-\infty} := P + \{-\infty\} & P_{-\infty} \times P_{-\infty} \supseteq \leq_{-\infty} := \leq \cup \{-\infty\} \times P_{-\infty} \\
P^{+\infty} := P + \{+\infty\} & P^{+\infty} \times P^{+\infty} \supseteq \leq^{+\infty} := \leq \cup P^{+\infty} \times \{+\infty\} \\
P_{-\infty}^{+\infty} := P + \{-\infty, +\infty\} & P_{-\infty}^{+\infty} \times P_{-\infty}^{+\infty} \supseteq \leq_{-\infty}^{+\infty} := \leq \cup \{-\infty\} \times P_{-\infty}^{+\infty} \cup P_{-\infty}^{+\infty} \times \{+\infty\}
\end{array}$$

Man beachte, dass $\leq_{-\infty}$, $\leq^{+\infty}$ und $\leq_{-\infty}^{+\infty}$ jeweils wieder lineare Ordnungen sind, falls $\leq$ eine lineare Ordnung war.

[†] Tatsächlich handelt es sich hier um Summen von Ordnungstypen, die weiter unten eingeführt werden.

Mit dem Begriff der Ordnung sind zwei weitere Begriffe eng verbunden: das *Minimum* und das *Maximum*.

Definition (Minimum, Maximum)**:** Sei $(P, \leq_P)$ eine partiell geordnete Menge und sei $Q \subseteq P$ eine beliebige Teilmenge von P.
Gibt es ein Element $\check{q} \in Q$, sodass $\check{q} \leq_P q'$ für alle Element $q' \in Q$ gilt, so heißt $\check{q}$ *Minimum* von Q bezüglich $\leq_P$, geschrieben als $\min_{\leq_P} Q = \check{q}$. Gibt es kein solches Element ist $\min_{\leq_P} Q$ undefiniert.
Gibt es ein Element $\hat{q} \in Q$, sodass $q' \leq_P \hat{q}$ für alle Elemente $q' \in Q$ gilt, so heißt $\hat{q}$ *Maximum* von Q bezüglich $\leq_P$, geschrieben als $\max_{\leq_P} Q = \hat{q}$. Gibt es kein solches Element ist $\max_{\leq_Q}$ undefiniert.

Aufgrund der Antisymmetrie partieller Ordnungen sind Minimum und Maximum eindeutig, falls sie existieren. Außerdem hat jede endliche linear geordnete Menge stets ein Minimum und ein Maximum.

Konvention: Ist die verwendete partielle Ordnung aus dem Zusammenhang klar, wird die Schreibweise zu $\min Q$ bzw $\max Q$ vereinfacht. Ist $\min Q$ oder $\max Q$ undefiniert, so wird ähnlich wie bei partiellen Funktionen $\max Q = \perp$ bzw $\min Q = \perp$ geschrieben.

Neben den verschiedenen Ordnungen haben auch andere besondere Relationen einen speziellen Namen. Der vielleicht bekannteste Fall einer solchen ist die *Äquivalenzrelation*.

Definition (Äquivalenzrelation)**:** Eine transitive, reflexive und symmetrische binäre Relation heißt *Äquivalenzrelation*.

Seien $(P_1, \leq_1)$ und $(P_2, \leq_2)$ zwei partiell geordnete Mengen. Sie haben den selben *Ordnungstyp*, wenn es eine ordnungserhaltende Bijektion zwischen ihnen gibt, d. h. wenn es eine Bijektion $\iota : P_1 \to P_2$ mit $p \leq_1 p' \iff \iota(p) \leq_2 \iota(p')$ für alle $p, p' \in P_1$ gibt. Der Ordnungstyp einer geordneten Menge $(P, \leq)$ bildet (wie man leicht nachrechnet) eine Klasse einer Äquivalenzrelation, bezeichnet als $\operatorname{ord}(P, \leq)$. Einige Ordnungstypen besitzen besondere Bezeichnungen:

Definition: ω bezeichnet den Ordnungstyp von $(\mathbb{N}, \leq)$, ω^* den von $(-\mathbb{N}, \leq)$[†]. Natürliche Zahlen $n \in \mathbb{N}$ bezeichnen den Ordnungstyp von $(\{1, 2, \dots, n\}, \leq)$.

Der Ordnungstyp einer linear geordneten Menge heißt *linearer* Ordnungstyp. Man überlegt sich leicht, dass diese Definition unabhängig vom gewählten Repräsentanten und damit wohldefiniert ist.

Konvention: Für einen Ordnungstypen μ bezeichnet $\leq_\mu$ im Folgenden die zugehörige Ordnung. Wird diese Notation verwendet, ist die Aussage von der Wahl des Repräsentanten von μ unabhängig oder der gewählte Repräsentant ergibt sich aus dem Kontext. Bei den oben genannten Ordnungstypen mit besonderen Bezeichnungen handelt es sich in der Regel dann um den Repräsentanten, über den sie dort definiert wurden.

Es folgen einige Konstruktionen mit partiell geordneten Mengen und ihren Ordnungstypen. Die erste dieser Konstruktionen wird auf eine einzelne partiell geordnete Menge angewendet.

[†]Es ist $\mathbb{N} = \{1, 2, \dots\}$ und $-\mathbb{N} = \{-1, -2, \dots\}$; die natürlichen Zahlen ab 1 zu definieren, wird sich im Folgenden als nützlich erweisen.

Definition (Duale Ordnung, dualer Ordnungstyp)**:** Sei $(P, \leq_P)$ eine partiell geordnete Menge. Definiere $-P := \{-p : p \in P\}$ als disjunkte Kopie von P. Definiere die duale partielle Ordnung $\leq_P^* := \{(-p_1, -p_2) : p_2 \leq_P p_1\}$. Sei $\mu = \operatorname{ord}(P, \leq_P)$ der Ordnungstyp der partiell geordneten Menge, dann bezeichnet $\mu^* := \operatorname{ord}(-P, \leq_P^*)$ den dualen Ordnungstyp zu μ.

Wieder gilt: Der duale Ordnungstyp ist vom gewählten Repräsentanten unabhängig. Dies zeigt eine einfache Rechnung. Ebenso durch einfache Rechnung zeigt sich, dass das Bilden des dualen Ordnungstyps eine Involution ist, also dass $(\mu^*)^* = \mu$ gilt.

Beispiel: ω^* ist der duale Ordnungstyp zu ω.

Lemma 2: Sei $n \in \mathbb{N}$ eine beliebige natürliche Zahl. Betrachtet man n als Ordnungstyp, so gilt: $n^* = n$.

Beweis. Definiere die offensichtlich bijektive Abbildung:

$$\begin{aligned} \iota : \{1, 2, \dots, n\} &\to \{-1, -2, \dots, -n\} \\ i &\mapsto i - n - 1 \end{aligned}$$

Ist $\leq$ die gebräuchliche „kleiner gleich"-Relation auf den natürlichen Zahlen, so entspricht $\leq^*$ der Restriktion der gebräuchlichen „kleiner gleich"-Relation auf den negativen Teil der ganzen Zahlen. Es ist also $i \leq j \iff \iota(i) \leq \iota(j)$ zu zeigen, dies gilt aber trivialerweise. $\square$

Die nächsten beiden Konstruktionen kombinieren zwei partiell geordnete Menge zu einer neuen.

Definition (Summe, Produkt partieller Ordnungen)**:** Seien $(P_1, \leq_1)$ und $(P_2, \leq_2)$ zwei partiell geordnete Mengen. Betrachte die beiden Mengen P_1 und P_2 als disjunkt und definiere $\leq_{1+2} \subseteq (P_1 + P_2) \times (P_1 + P_2)$ und $\leq_{1*2} \subseteq (P_1 \times P_2) \times (P_1 \times P_2)$ als

$$\begin{aligned} &\leq_{1+2} := \leq_1 \cup \leq_2 \cup (P_1 \times P_2) \text{ und} \\ &\leq_{1*2} := \{(p_1, p_2, p_1', p_2') : p_2 <_2 p_2'\} \cup \{(p_1, p_2, p_1', p_2') : p_2 = p_2' \text{ und } p_1 \leq_1 p_1'\}. \end{aligned}$$

$(P_1 + P_2, \leq_{1+2})$ ist die *Summe* von $(P_1, \leq_1)$ und $(P_2, \leq_2)$ und $(P_1 \times P_2, \leq_{1*2})$ das *Produkt*.

In der Summe sind die Elemente von P_1 also alle kleiner als die von P_2, wobei P_1 und P_2 selbst entsprechend der jeweiligen partiellen Ordnung geordnet sind. Beim Produkt wird zuerst entsprechend $\leq_2$ verglichen und bei Gleichheit entsprechend $\leq_1$. Die Summe und das Produkt linear geordneter Mengen sind selbst wieder linear geordnete Mengen; dies ergibt sich direkt aus der Definition.

Die Verknüpfung zweier partiell geordneter Mengen zu einer Summe oder einem Produkt überträgt sich auf die entsprechenden Ordnungstypen, da das Ergebnis unabhängig vom gewählten Repräsentanten ist. Das fasst das folgende Lemma zusammen, dessen Beweis hier nur skizziert werden soll:[†]

[†] Mehr Details finden sich zum Beispiel in [13].

Lemma: Seien $(P_1, \leq_{P_1})$ und $(Q_1, \leq_{Q_1})$ sowie $(P_2, \leq_{P_2})$ und $(Q_2, \leq_{Q_2})$ je zwei partiell geordnete Mengen vom selben Ordnungstyp. Seien ferner $(P_1 + P_2, \leq_{P,1+2})$ die Summe von $(P_1, \leq_{P_1})$ und $(P_2, \leq_{P_2})$ und $(Q_1 + Q_2, \leq_{Q,1+2})$ die von $(Q_1, \leq_{Q_1})$ und $(Q_2, \leq_{Q_2})$. Die analogen Produkte seien $(P_1 \times P_2, \leq_{P,1*2})$ und $(Q_1 \times Q_2, \leq_{Q,1*2})$
Dann gilt:

$$\begin{aligned} \operatorname{ord}(P_1 + P_2, \leq_{P,1+2}) &= \operatorname{ord}(Q_1 + Q_2, \leq_{Q,1+2}) \text{ und} \\ \operatorname{ord}(P_1 \times P_2, \leq_{P,1*2}) &= \operatorname{ord}(Q_1 \times Q_2, \leq_{Q,1*2}) \,. \end{aligned}$$

Beweisskizze. Sei $\iota_1 : P_1 \to Q_1$ eine der ordnungserhaltenden Bijektionen zwischen P_1 und Q_1, sei $\iota_2 : P_2 \to Q_2$ entsprechend eine der ordnungserhaltenden Bijektionen zwischen P_2 und Q_2. Definiere die ordnungserhaltende Bijektion

$$\begin{aligned} \iota_{1+2} : P_1 + P_2 &\to Q_1 + Q_2 \\ s &\mapsto \begin{cases} \iota_1(s) & \text{falls } p \in P_1 \\ \iota_2(s) & \text{falls } p \in P_2. \end{cases} \end{aligned}$$

Dass es sich bei ι_{1+2} tatsächlich um die gewünschte ordnungserhaltende Bijketion handelt, sieht man leicht durch Rechnung ein.

Für die Aussage über Produkte definiere

$$\begin{aligned} \iota_{1*2} : S_1 \times S_2 &\to T_1 \times T_2 \\ (s_1, s_2) &\mapsto (\iota_1(s_1), \iota_2(s_2)) \,. \end{aligned}$$

Auch hier zeigt sich durch eine Rechnung, dass es sich um eine ordnungserhaltende Bijketion handelt. □

Mit diesen Erkenntnissen lässt sich die Schreibweise vereinfachen. Für zwei Ordnungstypen $\mu = \operatorname{ord}(P, \leq_P)$ und $\nu = \operatorname{ord}(Q, \leq_Q)$ ist $\mu + \nu$ der von den Repräsentanten unabhängige Ordnungstyp $\operatorname{ord}(P + Q, \leq_{P+Q})$. Analog ist $\mu \cdot \nu$ der Ordnungstyp $\operatorname{ord}(P \times Q, \leq_{P*Q})$. Um auch komplexere Ausdrücke aus Summen und Produkten von Ordnungstypen schreiben zu können, wird – wie üblich – vereinbart, dass Produkte höhere Präzedenz haben als Summen. Außerdem können Klammer weggelassen werden, da die Operationen assoziativ sind, wie man durch einfache Rechnung leicht einsieht.†

Neben der Assoziativität von Summe und Produkt gelten auch einige Rechenregeln, insbesondere wenn man natürliche Zahlen als Ordnungstypen betrachtet.

Lemma 3: Seien n und m natürliche Zahlen. Dann gilt:

$$\begin{aligned} \operatorname{ord}(\{1, 2, \ldots, n\}, \leq) + \operatorname{ord}(\{1, 2, \ldots, m\}, \leq) &= \operatorname{ord}(\{1, 2, \ldots, n + m\}, \leq) \\ \operatorname{ord}(\{1, 2, \ldots, n\}, \leq) \cdot \operatorname{ord}(\{1, 2, \ldots, m\}, \leq) &= \operatorname{ord}(\{1, 2, \ldots, n \cdot m\}, \leq) \end{aligned}$$

Beweis. Die Aussagen sind eine direkte Folge aus der Eindeutigkeit endlicher linearer Ordnungen (vgl. dazu [13, S. 17]). □

†Mehr zur Assoziativität von Summe und Produkt findet sich wieder z. B. in [13]

1.1.6 Wörter

Sei Σ ein beliebiges Alphabet. Wie bereits erwähnt bezeichnet Σ^* die Menge aller endlichen Wörter über Σ. Mit Σ^+ soll die Menge aller nicht-leeren Wörter auf Σ^* bezeichnet werden. Ein Wort $w \in \Sigma^+$ ist damit eine endliche Sequenz aus Buchstaben $w = a_1 a_2 \dots a_n$ mit $a_i \in \Sigma$ für $i = 1, 2, \dots, n$. w lässt sich allerdings auch als Abbildung $w : \{1, 2, \dots, n\} \to \Sigma$ von Positionen im Wort in das Alphabet betrachten. Für endliche Wörter ist die Menge der Positionen ebenfalls endlich. Tatsächlich ist diese Menge zusammen mit der $\leq$-Ordnung auf den natürlichen Zahlen eine linear geordnete Menge vom Ordnungstyp n, wobei $n = |w| \in \mathbb{N}$ die Länge des Wortes w ist.

Mit dieser Betrachtungsweise ist es möglich den Begriff des Wortes auf eine unendliche linear geordnete Menge zu erweitern:

Definition (Verallgemeinertes Wort)**:** Sei $(P, \leq)$ eine linear geordnete Menge. Eine Abbildung $w : P \to \Sigma$ heißt (verallgemeinertes) *Wort* über Σ vom Ordnungstyp $\operatorname{ord}(P, \leq)$. Das Bild von w heißt auch *effektives Alphabet* von w, geschrieben als $\alpha(w) := w(P)$. Die Menge aller verallgemeinerten Wörter über Σ wird mit Σ^{LO} bezeichnet.

Konvention: Im Folgenden wird bei Wörtern nicht mehr zwischen linear geordneten Mengen vom selben Ordnungstyp unterschieden, d. h. für zwei linear geordnete Mengen $(P, \leq_P)$ und $(Q, \leq_Q)$, für die es eine ordnungserhaltende Bijektion $\iota : P \to Q$ gibt, wird ein Wort $w : Q \to \Sigma$ als gleich zum Wort $w' : P \to \Sigma$ mit $p \mapsto w(\iota(p))$ betrachtet.

Beispiel:

$$\begin{aligned} w : \mathbb{N} &\to \{a, b\} \\ i &\mapsto \begin{cases} a & \text{falls } i \text{ gerade} \\ b & \text{sonst} \end{cases} \end{aligned}$$

w ist ein verallgemeinertes Wort vom Ordnungstyp ω. Anschaulich besteht w aus einer nach rechts unendlich langen Abfolge von ab.

Wie endliche Wörter lassen sich auch verallgemeinerte Wörter verknüpfen.

Definition (Konkatenation verallgemeinerter Wörter)**:** Sei u ein Wort über Σ vom Ordnungstyp $\operatorname{ord}(P, \leq_P)$ und v eines vom Ordnungstyp $\operatorname{ord}(Q, \leq_Q)$. Dann ist uv ein Wort über Σ vom Ordnungstyp $\operatorname{ord}(P, \leq_P) + \operatorname{ord}(Q, \leq_Q)$ und definiert über

$$\begin{aligned} uv : P + Q &\to \Sigma \\ i &\mapsto \begin{cases} u(i) & \text{falls } i \in P \\ v(i) & \text{falls } i \in Q \end{cases} \end{aligned}$$

Die Assoziativität der Summe von Ordnungstypen überträgt sich direkt auf die Konkatenation verallgemeinerter Wörter, d. h. es gilt $(uv)w = u(vw)$. Außerdem stimmt die Konkatenation zweier endlicher Wörter betrachtet als verallgemeinerte Wörter mit der herkömmlichen Konkatenation überein, wie das folgende Lemma festhält. Auf einen formellen Beweis der Aussage soll hier allerdings verzichtet werden, da er sich sehr leicht ergibt.

Lemma 4: Seien $w_1 = a_1 a_2 \dots a_n$ und $w_2 = b_1 b_2 \dots b_m$ zwei endliche Wörter über dem Alphabet Σ. Sei $w' = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$ die herkömmliche Konkatenation von w_1 und w_2. Dann gilt:

$$w_1 w_2 = w'$$

Verknüpft man zwei verallgemeinerte Wörter w_1 und w_2, so ist das effektive Alphabet dieses Wortes die Vereinigung der effektiven Alphabete der beiden Ursprungswörter.

Eine weitere Form der Verknüpfung verallgemeinerter Wörter liefert die folgende Definition:

Definition (Potenz verallgemeinerter Wörter)**:** Sei w ein Wort über Σ vom Ordnungstyp $\operatorname{ord}(P, \leq_P)$ und sei $\mu = \operatorname{ord}(Q, \leq_Q)$ ein linearer Ordnungstyp. Dann ist w^μ ein Wort über Σ vom Ordnungstyp $\operatorname{ord}(P, \leq_P) \cdot \mu$ und definiert über

$$\begin{aligned} w^\mu : P \times Q &\to \Sigma \\ (i, q) &\mapsto w(i) \end{aligned}$$

Analog wie bei der Konkatenation überträgt sich bei der Potenz die Assoziativität des Produkts von Ordnungstypen, d. h. es gilt $(w^\mu)^\nu = w^{(\mu \cdot \nu)}$. Das effektive Alphabet eines Wortes ändert sich nach dieser Definition durch Potenzierung nicht.

Zwischen Konkatenation und Potenz verallgemeinerter Wörter besteht ein Zusammenhang ähnlich dem bei endlichen Wörtern:

Lemma 5: Sei w ein (verallgemeinertes) Wort über dem Alphabet Σ und n eine natürliche Zahl. Es ist

$$w^n = \underbrace{ww \dots w}_{n \text{ mal}}.$$

Insbesondere ist w^n ein endliches Wort, falls w endlich ist.

Auch der Beweis dieses Lemmas ergibt sich sehr leicht und soll daher hier nicht explizit aufgeführt werden.

1.2 π-Terme und Gleichungen

Bisher wurden zumeist nur weitgehend bekannte Begriffe und Konzepte behandelt. Auch in diesem Abschnitt wird sich viel Vertrautes finden lassen. Allerdings befinden sich darunter auch Definition und erste Erkenntnisse, die für das Weitere von so zentraler Bedeutung sind, dass sie einen eigenen Abschnitt verdienen.

1.2.1 π-Terme

Beim Untersuchen von Halbgruppen spielen π-Terme oft eine wichtige Rolle, da sie Elemente einer Halbgruppe und die zugehörigen Idempotenten gut beschreiben können. In der Regel werden sie dabei allerdings nicht π-Terme, sondern ω-Terme genannt. Diese Terminologie wird zwar auch im Titel dieser Arbeit verwendet, hat aber einen Nachteil: Der Buchstabe ω wird nicht nur bei ω-Termen, sondern auch zur Beschreibung des Ordnungstyps der natürlichen Zahlen verwendet. Außerdem bezeichnet ω oft auch noch die

Potenz aus Lemma 1. Aufgrund dieser Doppeldeutigkeit ist hier von nun an konsequent von π-Termen die Rede[†].

Doch worum genau handelt es sich bei einem π-Term nun? Am Anfang der Antwort auf diese Frage steht die syntaktischen Definition eines π-Terms:

Definition (π-Term über Σ)**:** Sei Σ ein beliebiges Alphabet. Ein *π-Term* über (dem Alphabet) Σ ist folgendermaßen definiert:

1. Jeder Buchstabe $a \in \Sigma$ ist ein π-Term über (dem Alphabet) Σ.
2. Sei w ein π-Term über Σ. Dann ist $(w)^\pi$ ein π-Term über Σ.
3. Seien u und v π-Terme über Σ. Dann ist (uv) ein π-Term über Σ

Konvention: Ist lediglich von einem π-Term statt von einem π-Term über Σ die Rede, ergibt sich das Alphabet aus dem Kontext.

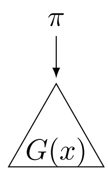
Konvention: Unnötige Klammern werden in π-Termen weggelassen, gemeint ist dann die Rechtsfaktorisierung des Terms. Dabei hat eine π-Potenz höhere Präzedenz als die Konkatenation.

Beispiel: $xy^\pi z = x((y)^\pi)z)$ ist ein π-Term über $\{x, y, z\}$.

Man beachte, dass nach der Definition eines π-Terms insbesondere jedes endliche, nichtleere Wort $w \in \Sigma^+$ ein π-Term ist! Tatsächlich handelt es sich bei π-Termen im Wesentlichen um Wörter mit der zusätzlichen Möglichkeit von π-Exponenten. Als solche besteht für sie die Möglichkeit sie graphisch über Syntax-Bäume darzustellen:

Definition (Syntax-Baum eines π-Terms)**:** Der *Syntax-Baum* $G(w)$ eines π-Terms w über Σ ist ein beschrifteter Baum, dessen Knoten linear geordnet sind. Er ist rekursiv definiert:

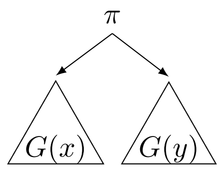
1. Ist der π-Term von der Form $w = a \in \Sigma$, also ein π-Term, der nur aus einem einzelnen Buchstaben besteht, so ist der Syntax-Baum definiert als $G(w) := (\{v_0\}, \emptyset, v_0, \lambda)$ mit $\lambda : \{v_0\} \to \Sigma + \{\pi, \cdot\}$ und $\lambda(v) := a$. Die zugehörige Ordnung der Knoten ist $\leq := \{(v_0, v_0)\}$.
2. Für einen π-Term von der Form $w = (x)^\pi$ mit $G(x) =: (V_x, E_x, v_{0,x}, \lambda_x)$ ist der Syntax-Baum durch $G(w) := (V, E_x \cup \{(v_{0,w}, v_{0,x})\}, v_{0,w}, \lambda)$ definiert, wobei $V := V_x + \{v_{0,w}\}$ und λ die Beschriftung λ_x um $\lambda(v_{0,w}) := \pi$ erweitert. Sei $\leq_x$ die Ordnung der Knoten von $G(x)$. Die Ordnung der Knoten von $G(w)$ ist dann $\leq := \leq_x \cup \{v_{0,w}\} \times V$.
   Graphisch:



3. Für einen π-Term von der Form $w = xy$ mit $G(x) := (V_x, E_x, v_{0,x}, \lambda_x)$ und $G(y) := (V_y, E_y, v_{0,y}, \lambda_y)$ ist der Syntax-Baum durch $G(w) := (V_x + V_y + \{v_{0,w}\}, E_x + E_y + \{(v_{0,w}, v_{0,x}), (v_{0,w}, v_{0,y})\}, v_{0,w}, \lambda)$ definiert. λ vereinigt dabei die Beschriftungen λ_x und λ_y und beschriftet $v_{0,w}$ mit $\cdot$. Die zugehörige Ordnung $\leq$ der Kno-

[†]Dies folgt der Notation in [11].

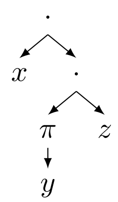
ten bildet die Summe der Ordnungen von $G(x)$ und $G(y)$ vereinigt mit $\{v_{0,w}\} \times (V_x + V_y + \{v_{0,w}\})$.
Graphisch:



Die Knoten mit Beschriftung π heißen *π-Knoten*, die mit Beschriftung $\cdot$ *Konkatenationsknoten*.

Man beachte, dass nach dieser Definition jeder Knoten aus dem Syntax-Baum eines π-Terms die Wurzel des Symtax-Baumes einen anderen π-Terms bildet; dieser kommt im ursprünglichen π-Term vor. Außerdem verifiziere man durch Nachrechnen, dass die Ordnung der Knoten eines π-Terms tatsächlich eine lineare Ordnung ist.

Beispiel: $xy^\pi z$ hat folgenden Syntax-Baum:



Bisher sind π-Terme rein syntaktisch definiert. Tatsächlich können die π-Exponenten als Platzhalter betrachten werden, deren eigentliche Bedeutung durch einen beliebigen Ordnungstyp interpretiert wird. Als Ergebnis dieses Vorgehens ergibt sich ein verallgemeinertes Wort.

Definition (Interpretation eines π-Terms)**:** Sei w ein π-Term über Σ und μ ein linearer Ordnungstyp. Das zugehörige verallgemeinerte Wort $[\![w]\!]_\mu$ ist folgendermaßen definiert:

- Ist $w = a \in \Sigma$, so ist $[\![w]\!]_\mu = a$.
- Ist w von der Form u^π, dann ist $[\![w]\!]_\mu := ([\![u]\!]_\mu)^\mu$.
- Ist w von der Form uv, dann ist $[\![w]\!]_\mu := [\![u]\!]_\mu [\![v]\!]_\mu$.

Ist w ein endliches Wort über Σ, so ist w nach Definition auch ein π-Term über Σ. Für jeden linearen Ordnungstyp μ ist nach dieser Definition dann $[\![w]\!]_\mu = w$. Tatsächlich ersetzt der Ordnungstyp im π-Term genau die π-Exponenten. Außerdem ist zu erwähnen, dass nach Lemma 5 und Lemma 4 $[\![w]\!]_n$ für einen π-Term w über dem Alphabet Σ und für jede natürliche Zahl n ein endliches Wort über Σ ist.

Die Ergänzung eines π-Terms um einen Ordnungstyp lässt sich in ähnlicher Weise auf den Syntax-Baum übertragen. Sei dazu $\mu = \operatorname{ord}(Q, \leq_Q)$ ein linearer Ordnungstyp. Betrachte einen Pfad p im Syntax-Baum $G(w) = (V, E, v_0, \lambda)$ eines π-Terms w über Σ, der an der Wurzel anfängt und an einem Blatt endet. p wird nun zu einem beschrifteten Pfad in $G(w)$ ergänzt. Dabei erhalten die π-Knoten eine Beschriftung aus Q und alle anderen Knoten die feste Beschriftung $\top$. Zur Vereinfachung wird ein solcher beschrifteter Pfad als *μ-Pfad* bezeichnet, dazu wird auch die Unterscheidung von einzelnen Repräsentanten des Ordnungstyps fallen gelassen. Alle μ-Pfade in $G(w)$ bilden die Menge $P_\mu(w)$.

Die μ-Pfade im Syntax-Baum eines π-Terms lassen sich linear ordnen, dazu ist die folgende Definition nützlich:

Definition: Sei w ein π-Term und μ ein linearer Ordnungstyp, seien außerdem p und p' zwei μ-Pfade in $G(w)$. Dann bezeichne $\delta_p(p')$ den ersten Knoten auf p, an dem sich p von p' unterscheidet.

Man beachte, dass $\delta_p(p')$ stets definiert ist.

Die eigentlich lineare Ordnung der μ-Pfade lautet dann wie folgt:

Definition: Seien p_1 und p_2 zwei μ-Pfade im Syntax-Baum $G(w) = (V, E, v_0, \lambda)$ eines π-Terms w über Σ, wobei $\mu = \mathrm{ord}\,(Q, \leq_Q)$ sei. Es ist $p_1 \leq_{\mu,w} p_2$ genau dann, wenn eine der folgenden Bedingungen eintritt:

- Es ist $p_1 = p_2$.
- Für $p_1 \neq p_2$ sei $v_1 = \delta_{p_1}(p_2)$ und $v_2 = \delta_{p_2}(p_1)$.
    - Es ist $v_1 = v_2$ und $p_1(v_1) \leq_Q p_2(v_2)$.
    - Es ist $v_1 < v_2$ (entsprechend der linearen Knotenordnung im Syntax-Baum).

Man beachte, dass μ-Pfade an Knoten, die keine π-Knoten sind, stets mit $\top$ beschriftet sind und sich so an diesen Knoten durch die Beschriftung nicht von anderen μ-Pfaden unterscheiden können. Gilt also $v_1 = v_2$, so muss dieser Knoten ein π-Knoten sein und damit sind $p_1(v_1)$ und $p_2(v_2)$ aus Q. Andererseits haben π-Knoten nach Konstruktion nur ein Kind, damit können v_1 und v_2, wenn sie unterschiedlich sind, nur Kinder eines Konkatenationsknotens sein. Schließlich zeigt einfaches Nachrechnen, dass $\leq_{\mu,w}$ tatsächlich eine lineare Ordnung ist.

Diese zunächst recht sperrig wirkende Definition der Ordnung der μ-Pfade ist in Wirklichkeit eng verwandt mit dem Wort $[\![w]\!]_\mu$:

Lemma 6: Sei w ein π-Term über Σ und $\mu = \mathrm{ord}\,(S, \leq_S)$ ein linearer Ordnungstyp. Der Ordnungstyp von $[\![w]\!]_\mu$ stimmt mit dem der μ-Pfade in $G(w)$ überein.

Beweis. Sei $\nu = \mathrm{ord}\,(T, \leq_T)$ der Ordnungstyp von $[\![w]\!]_\mu$. Der Beweis der Aussage erfolgt per Induktion über den Aufbau von w.

Ist $w = a \in \Sigma$ ein einzelner Buchstabe, so ist $\nu = 1$. In $G(w)$ gibt es dann genau einen Pfad, bestehend nur aus der Wurzel. Dieser bildet auch den einzigen μ-Pfad. Damit ist $\mathrm{ord}\,(P_\mu(w), \leq_{\mu,w}) = 1 = \nu$.

Ist $w = xy$, so sei $\nu_x = \mathrm{ord}\,(T_x, \leq_{T_x})$ der Ordnungstyp von $[\![x]\!]_\mu$ und $\nu_y = \mathrm{ord}\,(T_y, \leq_{T_y})$ der von $[\![y]\!]_\mu$. Nach Definition ist $[\![w]\!]_\mu = [\![x]\!]_\mu[\![y]\!]_\mu$ und damit $\nu = \nu_x + \nu_y$ bzw. $T = T_x + T_y$ und nach Induktion gibt es ordnungserhaltende Bijektionen $\sigma_x : P_\mu(x) \to T_x$ und $\sigma_y : P_\mu(y) \to T_y$. Diese lassen sich zu einer ordnungserhaltenden Bijektion

$$\begin{aligned} \sigma : P_\mu(w) &\to T \\ p = {}^{r}/{\top} \to p' &\mapsto \begin{cases} \sigma_x(p') & \text{falls } (p')_1 \text{ Knoten in } G(x) \text{ ist} \\ \sigma_y(p') & \text{falls } (p')_1 \text{ Knoten in } G(y) \text{ ist} \end{cases} \end{aligned}$$

erweitern. Da jeder μ-Pfad p in $G(w)$ an der Wurzel r anfängt und diese hier ein Konkatenationsknoten ist, ist $p(r) = \top$. Der zweite Knoten kann nach Konstruktion des

Syntax-Baumes dann nur entweder aus $G(x)$ oder aus $G(y)$ sein. Dies zeigt, dass σ wohldefiniert und total ist. Außerdem ergibt sich die Umkehrfunktion σ^{-1} durch Anwenden von σ_x^{-1} für Elemente aus T_x bzw. σ_y^{-1} für Elemente aus T_y und Erweitern des entstandenen Pfades p um $p(r) := \top$, wobei r die Wurzel von $G(w)$ sei. Um einzusehen, dass σ ordnungserhaltend ist, seien $p_1, p_2 \in P_\mu(w)$ mit $p_1 \leq_{\mu,w} p_2$. Dafür gibt es drei Möglichkeiten: Ist $p_1 = p_2$, so ist $\sigma(p_1) \leq_T \sigma(p_2)$ offensichtlich erfüllt, da $\leq_T$ reflexiv ist. Ist $p_1 \neq p_2$, so sei $v_1 = \delta_{p_1}(p_2)$ und $v_2 = \delta_{p_2}(p_1)$. v_1 und v_2 können nicht mit der Wurzel von $G(w)$ übereinstimmen, da beide Pfade dort gleich beschriftet sind. Liegen v_1 und v_2 beide in $G(x)$ oder beide in $G(y)$, so überträgt sich die Ordnungserhaltung von σ_x bzw. σ_y auf σ. Liegt einer in $G(x)$ und einer in $G(y)$, so kann es sich jeweils nur um die Wurzel des entsprechenden Syntax-Baumes handeln, da sonst vorher bereits eine Unterscheidung aufgetreten wäre. Aufgrund der Knotenordnung ist dann v_1 die Wurzel von $G(x)$ und v_2 die von $G(y)$ (es muss $v_1 < v_2$ gelten, weil $p_1 \leq_{\mu,w} p_2$ und $v_1 \neq v_2$ gilt). Damit wird p_1 auf einen Wert aus T_x und p_2 auf einen Wert aus T_y abgebildet. Nach Definition der Summe von Ordnungstypen gilt dann $\sigma(p_1) \leq_T \sigma(p_2)$. Umgekehrt gelte nun $\sigma(p_1) \leq_T \sigma(p_2)$ für zwei Pfade $p_1, p_2 \in P_\mu(w)$. Sind $\sigma(p_1)$ und $\sigma(p_2)$ beide aus T_x oder beide aus T_y, so überträgt sich auch in dieser Richtung die Ordnungserhaltung von σ_x bzw. σ_y. Ist eines aus T_x und eines aus T_y, muss nach Definition der Summe von Ordnungstypen $\sigma(p_1) \in T_x$ und $\sigma(p_2) \in T_y$ gelten. Dann muss aber $(p_1)_2$ aus $G(x)$ und $(p_2)_2$ aus $G(y)$ sein, genauer gesagt muss es sich um die jeweiligen Wurzeln der Syntax-Bäume handeln. Nach Definition ist dann $(p_1)_2 < (p_2)_2$ und damit $p_1 \leq_{\mu,w} p_2$.

Ist $w = (x)^\pi$, so sei $\nu_x = \operatorname{ord}(T_x, \leq_{T_x})$ der Ordnungstyp von $[\![x]\!]_\mu$. Nach Definition ist $[\![w]\!]_\mu = ([\![x]\!]_\mu)^\mu$ und $\nu = \nu_x \cdot \mu$ bzw. $T = T_x \times S$ und nach Induktion gibt es eine ordnungserhaltende Bijektion $\sigma_x : P_\mu(x) \to T_x$. Definiere:

$$\begin{aligned} \sigma : P_\mu(w) &\to T_x \times S \\ p = {}^r/_s \to p' &\mapsto (\sigma_x(p'), s) \end{aligned}$$

Man beachte, dass ein μ-Pfad in $G(w)$ an der Wurzel r beginnt und dort mit einem Element s aus S beschriftet ist. Daher ist die Definition von σ wohldefiniert und total. Zur Umkehrung lässt sich σ_x^{-1} auf das erste Element des Tupels anwenden und der entstehende Pfad um die Wurzel r mit dem zweiten Element als Beschriftung ergänzen. σ ist also tatsächlich eine Bijektion. Außerdem ist σ ordnungserhaltend: Seien $p_1 := {}^r/_{s_1} \to p'_1$ und $p_2 := {}^r/_{s_2} \to p'_2$ zwei beliebige μ-Pfade in $G(w)$ mit $p_1 \leq_{\mu,w} p_2$. Es gilt zwei Fälle zu unterscheiden: Gilt $s_1 = s_2$, so muss auch $p'_1 \leq_{\mu,x} p'_2$ gelten. Aufgrund der Ordnungserhaltung von σ_x und der Definition des Produkts von Ordnungstypen gilt dann $\sigma(p_1) = (\sigma_x(p'_1), s_1) \leq_T (\sigma_x(p'_2), s_1) = \sigma(p_2)$. Gilt andererseits $s_1 <_S s_2$, so ist $\sigma(p_1) \leq_T \sigma(p_2)$ entsprechend der Definition von $\nu = \nu_x \cdot \mu$ bzw. $\leq_T$. Für die Rückrichtung seien $p_1 := {}^r/_{s_1} \to p'_1$ und $p_2 := {}^r/_{s_2} \to p'_2$ wieder zwei beliebige μ-Pfade in $G(w)$, aber diesmal mit $(\sigma_x(p'_1), s_1) = \sigma(p_1) \leq_T \sigma(p_2) = (\sigma_x(p'_2), s_2)$. Nach Definition von $\leq_T$ ist $s_1 = s_2$ und $\sigma_x(p'_1) \leq_{T_x} \sigma_x(p'_2)$ oder $s_1 <_S s_2$. Im ersten Fall gilt $p'_1 \leq_{\mu,x} p'_2$ und wegen $s_1 = s_2$ somit $p_1 \leq_{\mu,w} p_2$. Im zweiten Fall unterscheiden sich die beiden Pfade p_1 und p_2 bereits durch die Beschriftung an der Wurzel von $G(w)$ und nach Definition gilt $p_1 \leq_{\mu,w} p_2$. $\square$

Dies Positionen eines Wortes $[\![w]\!]_\mu$ lassen sich also mit μ-Pfaden in $G(w)$ identifizieren. Diese Anschauung wird sich im Folgenden als nützlich erweisen.

1.2.2 Gleichungen

Gleichungen erlauben es Aussagen in kompakter Form auszudrücken. Dieser Unterabschnitt wird eine besondere Art von Gleichungen einführen, nämlich Gleichungen über Halbgruppen oder über Mengen von Halbgruppen. Tatsächlich ist es möglich diese auch über unendlichen Halbgruppen zu betrachten, der Einfachheit halber sollen hier aber nur solche über endlichen Halbgruppen betrachtet werden.

Zunächst soll formal definiert werden, was unter eine Gleichung über Halbgruppen zu verstehen ist:

Definition (Gleichung über einer Halbgruppe, Belegung)**:** Seien w_1 und w_2 zwei π-Terme über dem selben Alphabet $\Sigma = \{x_1, x_2, \ldots, x_n\}$. Dann heißt $w_1 = w_2$ *Gleichung* und die Buchstaben aus Σ auch *Variablen*. Außerdem heißt eine Abbildung $\sigma : \Sigma \to S$ von Σ in die Halbgruppe S *Belegung* der Variablen.

Konvention: Eine Belegung $\sigma : \Sigma \to S$ lässt sich zur Abbildung

$$\begin{aligned} \sigma' : \Sigma^+ &\to S \\ x_{i_1} x_{i_2} \ldots x_{i_m} &\mapsto \sigma(x_{i_1})\sigma(x_{i_2}) \ldots \sigma(x_{i_m}) \end{aligned}$$

erweitern. Im Folgenden wird eine Belegung $\sigma : \Sigma \to S$ daher stets auch als Abbildung $\sigma : \Sigma^+ \to S$ oder bei Monoiden als $\sigma : \Sigma^* \to M$ betrachtet.

Die Verwendung von Gleichungen wird erst sinnvoll, wenn ihnen ein Wahrheitswert zugeordnet werden kann. Für endliche Halbgruppen erfolgt dies in der nächsten Definition und für Mengen endlicher Halbgruppen in der darauf folgenden.

Definition (Wahrheitswert einer Gleichung 1)**:** Sei $w_1 = w_2$ eine Gleichung mit Variablen aus Σ und sei $\sigma : \Sigma \to S$ eine Belegung dieser Variablen mit Werten aus einer endlichen Halbgruppe S.
Die Gleichung $w_1 = w_2$ *gilt* in S *unter der Belegung* σ, falls $\sigma([\![w_1]\!]_{S!}) = \sigma([\![w_2]\!]_{S!})$ ist.
Die Gleichung $w_1 = w_2$ *gilt* in S, falls sie unter jeder Belegung $\sigma : \Sigma \to S$ gilt. Ansonsten *gilt* die Gleichung in S *nicht*.

Definition (Wahrheitswert einer Gleichung 2)**:** Sei N eine Menge von endlichen Halbgruppen. Eine Gleichung $w_1 = w_2$ *gilt* in N, falls sie in jeder Halbgruppe aus N gilt.

Um die Gültigkeit einer Gleichung zu beweisen kann das folgende Lemma sehr nützlich sein:

Lemma 7: Sie w ein π-Term über dem Alphabet Σ von Variablen und sei $\sigma : \Sigma \to S$ eine Belegung dieser Variablen mit Werten aus einer endlichen Halbgruppe S. Sei $c \in \mathbb{N}$ eine beliebige natürliche Zahl.
Dann gilt:

$$\sigma([\![w]\!]_{S!}) = \sigma([\![w]\!]_{c\cdot(S!)})$$

Beweis. Der Beweis erfolgt durch Induktion über den Aufbau von w. Ist $w = x \in \Sigma$ ein einzelner Buchstabe, gilt die Aussage direkt. Ist $w = w_1 w_2$, so ist nach Induktion $\sigma([\![w_i]\!]_{S!}) = \sigma([\![w_i]\!]_{c\cdot(S!)})$ für $i \in \{1,2\}$ und damit $\sigma([\![w]\!]_{S!}) = \sigma([\![w_1 w_2]\!]_{S!}) = \sigma([\![w_1]\!]_{S!}[\![w_2]\!]_{S!}) = \sigma([\![w_1]\!]_{S!})\sigma([\![w_2]\!]_{S!}) = \sigma([\![w_1]\!]_{c\cdot(S!)})\sigma([\![w_2]\!]_{c\cdot(S!)}) = \sigma([\![w]\!]_{c\cdot(S!)})$. Ist schließlich $w = (w')^\pi$, so ist nach Induktion $\sigma([\![w']\!]_{S!}) = \sigma([\![w']\!]_{c\cdot(S!)})$. Also ist:

$$\begin{aligned}
\sigma([\![w]\!]_{S!}) &= \sigma\left(([\![w']\!]_{S!})^{S!}\right) && \text{(Definition)}\\
&= \sigma\left([\![w']\!]_{c\cdot(S!)}\right)^{S!} && \text{(Lemma 5, Homomorphismus)}\\
&= \sigma\left([\![w']\!]_{c\cdot(S!)}\right)^{c\cdot(S!)} && \text{(Wahl von } S!)\\
&= \sigma\left(([\![w']\!]_{c\cdot(S!)})^{c\cdot(S!)}\right) && \text{(Lemma 5, Homomorphismus)}\\
&= \sigma([\![w]\!]_{c\cdot(S!)})
\end{aligned}$$

□

Die Definition der Wahrheitswerte von Gleichungen erlaubt es nun das Wortproblem für π-Terme, das zentrale Element der Betrachtungen dieser Arbeit zu definieren.

Definition (Das Wortproblem für π-Terme)**:** Das Wortproblem für π-Terme einer Menge N von endlichen Halbgruppen ist das folgende Entscheidungsproblem:
Eingabe: Zwei π-Terme w_1 und w_2 über dem selben Alphabet.
Frage: Gilt die Gleichung $w_1 = w_2$ in N?

1.3 Ranker und die Trotter-Weil-Hierarchie

1.3.1 Ranker

In [14] führen Schwentick, Thérien, und Vollmer sogenannte *turtle programs* als technisches Hilfsmittel ein, weisen aber auch darauf hin, dass sich diese als nützliches Werkzeug in einem weiteren Umfeld erweisen könnten. Tatsächlich griffen Weis und Immerman in [19] das Konzept unter dem Namen *ranker* auf. Ebendort finden sich Ansätze zu *condensed rankers*, wie sie in [8] genannt werden. Kondensierte Ranker haben auch Verbindungen zur eindeutigen Intervall-Temporallogik von Lodaya, Pandya und Shah [9]. Durch diese Verbindungen sind Ranker nützliche bei der Untersuchung von zwei Variablenlogik, wie im nächsten Unterabschnitt ausgeführt ist. Dieser Unterabschnitt führt Ranker zunächst im Zusammenhang mit den in Unterabschnitt 1.1.6 besprochenen verallgemeinerten Wörtern ein.

Definition (Ranker)**:** Sei Σ ein beliebiges Alphabet. Definiere $X_\Sigma := \{X_a : a \in \Sigma\}$, $Y_\Sigma = \{Y_a : a \in \Sigma\}$ und $Z_\Sigma = X_\Sigma \cup Y_\Sigma$. Ein nicht-leeres endliches Wort aus Z_Σ^+ heißt *Ranker* über Σ.
Sei w ein (verallgemeinertes) Wort vom Ordnungstyp $\operatorname{ord}(P, \leq)$ über Σ. Definiere für

$a \in \Sigma$ die partiellen Abbildungen:

$$\begin{aligned} {}^{w}X_a : P_{-\infty} &\to_p P \\ p &\mapsto \min_{\leq_{-\infty}}\{p' : w(p') = a \text{ und } p <_{-\infty} p'\} \end{aligned}$$

$$\begin{aligned} {}^{w}Y_a : P^{+\infty} &\to_p P \\ p &\mapsto \max_{\leq^{+\infty}}\{p' : w(p') = a \text{ und } p' <^{+\infty} p\} \end{aligned}$$

Zur Vereinfachung sei ${}^{w}X_a(\pm\infty) := {}^{w}X_a(-\infty)$ und ${}^{w}Y_a(\pm\infty) := {}^{w}Y_a(+\infty)$. Für einen Ranker der Länge ≥ 2 definiere rekursiv für alle $a \in \Sigma$:

$$\begin{aligned} {}^{w}rX_a : \{\pm\infty\} + P &\to_p P \\ p &\mapsto {}^{w}X_a({}^{w}r(p)) \end{aligned}$$

und

$$\begin{aligned} {}^{w}rY_a : \{\pm\infty\} + P &\to_p P \\ p &\mapsto {}^{w}Y_a({}^{w}r(p)) \end{aligned}$$

Definiere:

$$r \models w :\Longleftrightarrow {}^{w}r(\pm\infty) \neq \perp$$

Ein Ranker besteht also aus Instruktionen der Form „gehe zum nächsten a“ oder „gehe zum letzten a“, die sich auf ein verallgemeinertes Wort anwenden lassen. Der Ranker ist dabei entweder undefiniert – z. B. ist ${}^{b}X_a(\pm\infty)$ undefiniert – oder er beschreibt eine Position im Wort. Dazu beachte man insbesondere, dass der Ranker nicht auf einen der Werte $-\infty$, $+\infty$ oder $\pm\infty$ abbilden kann, was auch für die vorherige Definition eine Rolle spielt. Er wird stets von links nach rechts ausgewertet. Es ist unmittelbar klar, dass, falls ${}^{w}r(p)$ für ein $p \in \{\pm\infty\} + P$ definiert ist, ${}^{w}r'(p)$ auch für jedes Präfix r' von r definiert ist.

Eine weitere Beobachtung ist im Umgang mit Rankern noch wichtig: Ein Ranker aus Z_Σ^+ lässt sich nicht nur auf Wörter deren effektives Alphabet Σ entspricht anwenden! Der Ranker X_a oder Y_a ist nach Definition einfach undefiniert, sollte der Buchstabe a im Wort nicht vorkommen.

Neben der allgemeinen Form von Rankern, wie sie gerade definiert wurde, gibt es eine andere Ranker-Art, die im weiteren Verlauf eine größere Rolle spielen wird.

Definition (Kondensierte Ranker)**:** Sei r ein Ranker über einem beliebigen Alphabet Σ und sei w ein (verallgemeinertes) Wort vom Ordnungstyp $\operatorname{ord}(P, \leq)$. Ein Element $(l, p, r) \in P_{-\infty} \times P \times P^{+\infty}$ heißt *zulässig*, falls $l <_{-\infty} p <^{+\infty} r$ gilt. Definiere die Menge der zulässigen Elemente als:

$$P^{\mathrm{c}} := \{(l, p, r) \in P_{-\infty} \times P \times P^{+\infty} : (l, p, r) \text{ zulässig}\}.$$

Definiere zudem für $a \in \Sigma$ die Abbildungen

$$\begin{aligned} {}^w{X_a}^{\mathrm{c}} : \{\pm\infty\} + P^{\mathrm{c}} &\to_p P^{\mathrm{c}} \\ \pm\infty &\mapsto (-\infty, {}^wX_a(-\infty), +\infty), \text{ falls } {}^wX_a(-\infty) \neq \bot \\ (l,p,r) &\mapsto (p, p', r) \text{ mit } p' := {}^wX_a(p), \\ & \text{ falls } p' \neq \bot \text{ und } (p,p',r) \text{ zulässig} \end{aligned}$$

und

$$\begin{aligned} {}^w{Y_a}^{\mathrm{c}} : \{\pm\infty\} + P^{\mathrm{c}} &\to_p P^{\mathrm{c}} \\ \pm\infty &\mapsto (-\infty, {}^wY_a(+\infty), +\infty), \text{ falls } {}^wY_a(+\infty) \neq \bot \\ (l,p,r) &\mapsto (l, p', p) \text{ mit } p' := {}^wY_a(p), \\ & \text{ falls } p' \neq \bot \text{ und } (l,p',p) \text{ zulässig} \end{aligned}$$

Für einen Ranker der Länge ≥ 2 definiere rekursiv für alle $a \in \Sigma$

$$\begin{aligned} {}^wr{X_a}^{\mathrm{c}} : \{\pm\infty\} + P^{\mathrm{c}} &\to_p P^{\mathrm{c}} \\ p_{\mathrm{c}} &\mapsto {}^w{X_a}^{\mathrm{c}}\left({}^wr^{\mathrm{c}}(p_{\mathrm{c}})\right) \end{aligned}$$

und

$$\begin{aligned} {}^wr{Y_a}^{\mathrm{c}} : \{\pm\infty\} + P^{\mathrm{c}} &\to_p P^{\mathrm{c}} \\ p_{\mathrm{c}} &\mapsto {}^w{Y_a}^{\mathrm{c}}\left({}^wr^{\mathrm{c}}(p_{\mathrm{c}})\right) \end{aligned}$$

Definiere:

$$r \models^{\mathrm{c}} w :\Longleftrightarrow {}^wr^{\mathrm{c}}(\pm\infty) \neq \bot$$

Konvention: Ist das Wort, auf das ein (kondensierter oder nicht-kondensierter) Ranker angewendet wird, aus dem Kontext ersichtlich, entfällt es als Index in der linken oberen Ecke.

Wie bei (normalen) Rankern ist für einen kondensierten Ranker r auch ${}^wr'^{\mathrm{c}}$ für jedes Präfix r' von r an allen Stellen definiert, an denen ${}^wr^{\mathrm{c}}$ definiert ist. Es sei an dieser Stelle außerdem hervorgehoben, dass sich aus der letzten Definition folgendes ergibt: Ist ${}^wr^{\mathrm{c}}(\pm\infty) = (l,p,r) \neq \bot$ definiert, so ist $p = {}^wr(\pm\infty)$ und damit insbesondere definiert. Genauso gilt: Ist ${}^wr^{\mathrm{c}}(l,p,r) = (l',p',r') \neq \bot$ definiert, so ist $p' = {}^wr(p)$ und damit insbesondere definiert. Die umgekehrte Richtung gilt allerdings im Allgemeinen nicht, da nicht alle vom Ranker erreichten Werte zu zulässigen Tupeln (l,p,r) führen müssen.

Weitere Zusammenhänge zwischen r und r^{c} ergeben sich durch die nächsten Lemmata:

Lemma 8: Sei $r = Z_1Z_2 \ldots Z_n$ ein Ranker über Σ und sei w ein Wort vom Ordnungstyp $\operatorname{ord}(P, \leq)$, sodass ${}^wr^{\mathrm{c}}(p_{\mathrm{c}})$ für $p_{\mathrm{c}} \in \{\pm\infty\} + P^{\mathrm{c}}$ definiert ist. Es sei $(l_i, p_i, r_i) := {}^wZ_1Z_2 \ldots {Z_i}^{\mathrm{c}}(p_{\mathrm{c}})$ für $1 \leq i \leq n$.
Dann gilt $l_1 \leq_{-\infty} l_2 \leq_{-\infty} \cdots \leq_{-\infty} l_n <_{-\infty} p_n <^{+\infty} r_n \leq^{+\infty} r_{n-1} \leq^{+\infty} \cdots \leq^{+\infty} r_1$.
Außerdem gilt $l_i <_{-\infty} l_{i+1}$ oder $r_{i+1} <^{+\infty} r_i$ für $1 \leq i \leq n-1$.

Beweis. Der Beweis erfolgt per Induktion über n. Für $n = 1$ gilt $l_1 = l_n <_{-\infty} p_n = p_1 <^{+\infty} r_n = r_1$, da (l_1, p_1, r_1) zulässig sein muss. Sei $n > 1$ und die Aussage für $Z_1 Z_2 \dots Z_{n-1}$ gezeigt. Ist $Z_n = X_a$ für ein $a \in \Sigma$, so ist $(l_n, p_n, r_n) = X_a^{\mathrm{c}}(l_{n-1}, p_{n-1}, r_{n-1})$ und damit $l_n = p_{n-1}$ und $r_n = r_{n-1}$. Es gilt $l_1 \leq_{-\infty} l_2 \leq_{-\infty} \cdots \leq_{-\infty} l_{n-1}$ nach Induktion, $l_{n-1} <_{-\infty} p_{n-1} = l_n <_{-\infty} p_n <^{+\infty} r_n = r_{n-1}$ nach Zulässigkeit aller (l_i, p_i, r_i) und $r_{n-1} \leq^{+\infty} r_{n-2} \leq^{+\infty} \cdots \leq^{+\infty} r_1$ ebenfalls nach Induktion. Ist $Z_n = Y_a$ für ein $a \in \Sigma$, so ist $(l_n, p_n, r_n) = Y_a^{\mathrm{c}}(l_{n-1}, p_{n-1}, r_{n-1})$ und damit $l_n = l_{n-1}$ und $r_n = p_{n-1}$. Es gilt wieder $l_1 \leq_{-\infty} l_2 \leq_{-\infty} \cdots \leq_{-\infty} l_{n-1}$ nach Induktion, $l_{n-1} = l_n <_{-\infty} p_n <^{+\infty} r_n = p_{n-1} <^{+\infty} r_{n-1}$ nach Zulässigkeit aller (l_i, p_i, r_i) und $r_{n-1} \leq^{+\infty} r_{n-2} \leq^{+\infty} \cdots \leq^{+\infty} r_1$ ebenfalls nach Induktion. □

Lemma: Sei $r = Z_1 Z_2 \dots Z_n$ ein Ranker über Σ und sei w ein Wort vom Ordnungstyp $\mathrm{ord}\,(P, \leq)$, sodass ${}^w r^{\mathrm{c}}(\pm\infty)$ definiert ist. Es sei $(l_i, p_i, r_i) := {}^w Z_1 Z_2 \dots Z_i{}^{\mathrm{c}}(\pm\infty)$ für $1 \leq i \leq n$.
Dann ist ${}^w r(\pm\infty)$ definiert und es gilt für jedes $1 \leq i \leq n$ entweder $\forall n \geq j > i : p_i < p_j$ oder $\forall n \geq j > i : p_j < p_i$.

Beweis. Wie bereits bemerkt, gilt: ${}^w Z_1 Z_2 \dots Z_i(\pm\infty) = p_i$ für alle $1 \leq i \leq n$, damit ist insbesondere ${}^w r(\pm\infty)$ definiert.

Für $i = n$ ist nichts zu zeigen. Sei also $1 \leq i < n$. Ist $Z_{i+1} = X_a$ für ein $a \in \Sigma$, dann ist $l_{i+1} = p_i$. Außerdem gilt nach Lemma 8 $p_i = l_{i+1} \leq l_{i+2} \leq \cdots \leq l_n$. Dazu beachte man, dass $l_{i+1} \neq -\infty$ und damit auch alle nachfolgenden $l_j \neq -\infty$ sind und somit $\leq$ statt $\leq_{-\infty}$ geschrieben werden kann. Aufgrund der Zulässigkeit aller (l_j, p_j, r_j) gilt dann $p_i < p_j$ für alle $n \geq j \geq i + 1$. Ist $Z_{i+1} = Y_a$ für ein $a \in \Sigma$, dann ist $r_{i+1} = p_i$. Wiederum nach Lemma 8 gilt $r_n \geq r_{n-1} \geq \cdots \geq r_{i+1} = p_i$ und damit $p_j > p_i$ für alle $n \geq j \geq i + 1$ nach der selben Argumentation. □

Lemma: Sei $r = Z_1 Z_2 \dots Z_n$ ein Ranker über Σ. Sei w ein Wort vom Ordnungstyp $\mathrm{ord}\,(P, \leq)$ und ${}^w r(\pm\infty)$ definiert. Sei $p_i := {}^w Z_1 Z_2 \dots Z_i(\pm\infty)$ für $1 \leq i \leq n$.
Gilt für alle $1 \leq i \leq n$ entweder $p_i < p_j$ oder $p_j < p_i$ für alle $n \geq j > i$, so ist ${}^w r^{\mathrm{c}}(\pm\infty)$ definiert.

Beweis. Der Beweis erfolgt per Induktion über den Aufbau von r. Für Z_1 gilt: $Z_1{}^{\mathrm{c}}(\pm\infty)$ ist definiert, falls $Z_1(\pm\infty)$ definiert ist.

Sei $Z_1 Z_2 \dots Z_i{}^{\mathrm{c}}(\pm\infty)$ für alle $i < k$ mit $k > 1$ definiert. Für $1 \leq i < k$ setze $(l_i, p_i, r_i) := Z_1 Z_2 \dots Z_i{}^{\mathrm{c}}(\pm\infty)$. Man beachte, dass dies kein Widerspruch zur Definition der p_i oben darstellt. Ist $Z_k = X_a$ für ein $a \in \Sigma$, so ist $Z_1 Z_2 \dots Z_{k-1} X_a{}^{\mathrm{c}}(\pm\infty) = X_a{}^{\mathrm{c}}(l_{k-1}, p_{k-1}, r_{k-1})$. Dies ist definiert, da zum einen $X_a(p_{k-1}) = p_k$ definiert ist und zum anderen (p_{k-1}, p_k, r_{k-1}) zulässig ist: Nach Definition von ${}^w X_a$ ist $p_{k-1} < p_k$, außerdem muss nach Definition der kondensierten Ranker $r_{k-1} = +\infty$ oder $r_{k-1} = p_{i_0}$ für ein $i_0 < k - 1$ sein. Im ersten Fall gilt $p_k <^{+\infty} r_{k-1}$, im zweiten ist $p_{k-1} < r_{k-1} = p_{i_0}$ wegen der Zulässigkeit von $(l_{k-1}, p_{k-1}, r_{k-1})$ und damit nach Voraussetzung des Lemmas $p_k < p_{i_0} = r_{k-1}$.

Ist $Z_k = Y_a$ für ein $a \in \Sigma$, so erfolgt die Argumentation weitgehend analog: Es ist dann $Z_1 Z_2 \dots Z_{k-1} Y_a{}^{\mathrm{c}}(\pm\infty) = Y_a{}^{\mathrm{c}}(l_{k-1}, p_{k-1}, r_{k-1})$. Dies ist definiert, da zum einen

$Y_a(p_{k-1}) = p_k$ definiert ist und zum anderen (l_{k-1}, p_k, p_{k-1}) zulässig ist: Nach Definition von ${}^wY_a$ ist $p_k < p_{k-1}$, außerdem muss nach Definition der kondensierten Ranker $l_{k-1} = -\infty$ oder $l_{k-1} = p_{i_0}$ für ein $i_0 < k-1$ sein. Im ersten Fall gilt $l_{k-1} <_{-\infty} p_k$, im zweiten ist $l_{k-1} = p_{i_0} < p_{k-1}$ wegen der Zulässigkeit von $(l_{k-1}, p_{k-1}, r_{k-1})$ und damit nach Voraussetzung des Lemmas $l_{k-1} = p_{i_0} < p_k$. $\square$

Tatsächlich ist damit die Äquivalenz der hier gegebenen Definition kondensierter Ranker mit der aus [7] gegeben (vgl. dazu auch [8]), falls man diese sinngemäß auf verallgemeinerte Wörter erweitert. Insbesondere sind die Definitionen aber auch auf endlichen Wörtern gleich.

Im Umfeld der Ranker sind noch weitere Begriffe wichtig.

Definition (Block, Unterscheidbarkeit unter Rankern)**:** Sei r ein Ranker über Σ. Ein maximaler Faktor von r aus $X_\Sigma^+ \cup Y_\Sigma^+$ heißt *Block* von r. $R_{m,n}$ ist die Menge aller Ranker der Länge $\leq n$ mit maximal m Blöcken über beliebigem Alphabet. R_m ist die Menge aller Ranker beliebiger Länge mit maximal m Blöcken über beliebigem Alphabet. Definiere:

$$\begin{aligned} R_{m,n}^X &:= (R_{m,n} \cap X_\Sigma Z_\Sigma^*) \cup R_{m-1,n} \\ R_{m,n}^Y &:= (R_{m,n} \cap Y_\Sigma Z_\Sigma^*) \cup R_{m-1,n} \\ R_m^X &:= (R_m \cap X_\Sigma Z_\Sigma^*) \cup R_{m-1} \\ R_m^Y &:= (R_m \cap Y_\Sigma Z_\Sigma^*) \cup R_{m-1} \end{aligned}$$

Definiere außerdem die folgenden Relationen über verallgemeinerten Wörtern:

$$\begin{aligned} u \equiv_{m,n} v &:\iff (\forall r \in R_{m,n} : r \models^{\mathrm{c}} u \iff r \models^{\mathrm{c}} v) \\ u \equiv_{m,n}^X v &:\iff \big(\forall r \in R_{m,n}^X : r \models^{\mathrm{c}} u \iff r \models^{\mathrm{c}} v\big) \\ u \equiv_{m,n}^Y v &:\iff \big(\forall r \in R_{m,n}^Y : r \models^{\mathrm{c}} u \iff r \models^{\mathrm{c}} v\big) \\ u \equiv_m v &:\iff (\forall r \in R_m : r \models^{\mathrm{c}} u \iff r \models^{\mathrm{c}} v) \\ u \equiv_m^X v &:\iff \big(\forall r \in R_m^X : r \models^{\mathrm{c}} u \iff r \models^{\mathrm{c}} v\big) \\ u \equiv_m^Y v &:\iff \big(\forall r \in R_m^Y : r \models^{\mathrm{c}} u \iff r \models^{\mathrm{c}} v\big) \end{aligned}$$

Die Relationen $\equiv_{m,n}$, $\equiv_{m,n}^X$ und $\equiv_{m,n}^Y$ sind über verallgemeinerten Wörtern definiert. Damit sind sie aber insbesondere auch Relationen über den endlichen Wörtern aus Σ^* für ein Alphabet Σ. Tatsächlich sind sie dort sogar Kongruenzen von endlichem Index (vgl. [7, Proposition 5.7]). Die Mengen $\Sigma^*/{\equiv_{m,n}}$, $\Sigma^*/{\equiv_{m,n}^X}$ und $\Sigma^*/{\equiv_{m,n}^Y}$ der Kongruenzklassen bilden daher jeweils ein endliches Monoid.†

1.3.2 Die Trotter-Weil-Hierarchie

Diese Arbeit beschäftigt sich mit dem Wortproblem für ω-Terme über Zweivariablenlogik. Es wurde bereits angeführt, dass ω-Terme hier tatsächlich als π-Terme bezeichnet werden. Es überrascht daher vielleicht weniger, dass auch die Zweivariablenlogik nur am Rande

†Für Details zu Kongruenzen und Monoiden aus Kongruenzklassen siehe z. B. [4, S. 22ff.]

auftaucht, obwohl sie eine zentrale Rolle spielt. Um dies zu verstehen ist es zunächst notwendig sich mit Prädikatenlogik erster Stufe $\mathbf{FO}[<]$ zu beschäftigen. Diese kann auf endliche Wörter angewendet werden: Dabei beziehen sich die Werte von Variablen auf Positionen im Wort und das spezielles Prädikat $\lambda(x) = a$ erlaubt es zu fordern, dass die durch x bestimmte Position mit a beschriftet ist. Weitere Prädikate der Form $x < y$ erlauben zusätzlich den Vergleich von Positionen. Es ist bekannt, dass es zu jedem Satz, also jeder Formel, die keine freien Variablen enthält, aus $\mathbf{FO}[<]$ bereits einen äquivalenten Satz gibt, der nur drei Variablen verwendet. Schränkt man die Anzahl der Variablen auf zwei ein, erhält man jene Logik $\mathbf{FO}^2[<]$, die im Titel mit Zweivariablenlogik bezeichnet wird. Ein Satz aus einer der Logiken definiert nun eine Sprache endlicher Wörter; diese besteht genau aus jenen Wörtern, die den Satz erfüllen. Wird eine Sprache endlicher Wörter durch einen Satz aus einer Logik definiert, so heißt sie in dieser Logik *definierbar*.

Es ist nun möglich Zusammenhänge zwischen in bestimmten Logiken definierbaren Sprachen und algebraischen Eigenschaften jener Monoide, die diese Sprachen erkennen, herzustellen. Der für diese Arbeit wichtigste Zusammenhang ist dabei der folgende: Eine Sprache ist in $\mathbf{FO}^2[<]$ definierbar genau dann, wenn sie durch ein Monoid in der Varietät $\mathbf{DA}$ erkannt wird [16, 3]. Mehr zu den hier beschrieben Logiken, einschließlich einer formalen Definition, findet sich z. B. in [3].

Bei der Trotter-Weil-Hierarchie handelt es sich nun um eine Hierarchie aus Varietäten, die Teilmengen von $\mathbf{DA}$ sind. Namensgebend ist hier die erste Untersuchung dieser Hierarchie durch Trotter und Weil [17]. Die Hierarchie besteht aus den Ecken $\boldsymbol{R_m}$ und $\boldsymbol{L_m}$, den $\cap$-Ebenen $\boldsymbol{R_m} \cap \boldsymbol{L_m}$ und den $\vee$-Ebenen $\boldsymbol{R_m} \vee \boldsymbol{L_m}$, alle jeweils für alle $m \in \mathbb{N}$. Die bekannten Varietäten der $\mathcal{R}$-trivialen und der $\mathcal{L}$-trivialen Monoide treten als Ecken in der Trotter-Weil-Hierarchie auf, die der $\mathcal{J}$-trivialen Monoide als $\cap$-Ebene. Im Folgenden wird das Wortproblem für π-Terme der Ecken und der $\vee$-Ebene genauer untersucht. Dabei soll auf eine genaue Definition der genannten Varietäten verzichtet werden, da sie zum einen viele Konzepte aus der Halbgruppentheorie benötigt und zum anderen für die weiteren Ergebnisse nicht notwendig ist.[†] Stattdessen werden die für das Weitere entscheidenden Eigenschaften hier in der Form von Sätzen vorgestellt, die sich auch als alternative, aber äquivalente, Definition betrachten lassen. Hierbei spielen Logiken und algebraische Eigenschaften nur implizit eine Rolle, wichtig sind dagegen die im letzten Unterabschnitt definierten Ranker.

Theorem 1: Sei Σ ein beliebiges Alphabet und M ein endliches Monoid. Sei ferner $\varphi : \Sigma^* \to M$ ein surjektiver Monoidhomomorphismus und $m \geq 1$.
Dann gilt:

- $M \in \boldsymbol{R_m} \iff \left(\forall u, v \in \Sigma^* : u \equiv^X_{m,(m+1)|M|-m} v \implies \varphi(u) = \varphi(v)\right)$.
- $M \in \boldsymbol{L_m} \iff \left(\forall u, v \in \Sigma^* : u \equiv^Y_{m,(m+1)|M|-m} v \implies \varphi(u) = \varphi(v)\right)$.
- $M \in \boldsymbol{R_m} \vee \boldsymbol{L_m} \iff (\exists n \geq 1 \forall u, v \in \Sigma^* : u \equiv_{m,n} v \implies \varphi(u) = \varphi(v))$.

Außerdem gilt:

- $\forall n \in \mathbb{N} : \Sigma^*/{\equiv^X_{m,n}} \in \boldsymbol{R_m}$

[†]Eine solche formale Definition findet sich dabei in [7]

- $\forall n \in \mathbb{N} : \Sigma^*/{\equiv^Y_{m,n}} \in \boldsymbol{L_m}$
- $\forall n \in \mathbb{N} : \Sigma^*/{\equiv_{m,n}} \in \boldsymbol{R_m} \vee \boldsymbol{L_m}$

Beweis. Der erste Teil entspricht Theorem 5.10 aus [7]; im Beweis dafür werden auch die ersten beiden Punkte des zweiten Teils gezeigt. Der dritte Punkt des zweiten Teils schließlich ist eine direkte Folge aus dem ersten Teil. □

Beim ersten Teil der Aussage wird später die Richtung von links nach rechts entscheidend sein. Deshalb sei hier darauf hingewiesen, dass diese Implikation für alle Homomorphismen (also nicht nur surjektive) gilt: Ist ein Homomorphismus φ nicht surjektiv erweitere ihn und das Alphabet Σ so, dass er es ist. Dies ist stets möglich (wie man sich leicht überlegt). Da die Aussage für alle endlichen Wörter über dem erweiterten Alphabet gilt, gilt sie natürlich auch für alle Wörter aus dem ursprünglichen Alphabet. Dort stimmt der Homomorphismus aber bereits mit seiner Erweiterung überein.

Schließlich soll noch eine weitere Eigenschaft der Trotter-Weil-Hierachie in einem Satz festgehalten werden. Dank dieser Eigenschaft lassen sich die meisten Ergebnisse zur Trotter-Weil-Hierarchie auf **DA** übertragen.

Theorem 2: Es ist:

$$\mathbf{DA} = \bigcup_{m \in \mathbb{N}} \boldsymbol{R_m} \vee \boldsymbol{L_m}$$

Beweis. Die Aussage ergibt sich aus Theorem 4.1 in [7]. □

2 Entscheidbarkeit

Nachdem nun alle benötigten Begriffe eingeführt sind, wird sich dieses Kapitel mit dem ersten zentralen Ergebnis dieser Arbeit beschäftigen: Die Entscheidbarkeit des Wortproblems für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie sowie von **DA**. Dazu werden zunächst Ranker im Kontext von π-Termen erneut diskutiert, anschließend wird eine Form der Normalisierung von $(\omega + \omega^*)$-Pfaden im Syntax-Baum eines π-Terms eingeführt, die dann die Entscheidbarkeit der Probleme zeigt.

2.1 Ranker und π-Terme

2.1.1 Technische Hilfsmittel

Lemma 6 hat gezeigt, dass sich die Positionen in einem Wort $[\![w]\!]_\mu$ als μ-Pfade im Syntax-Baum $G(w)$ eines π-Terms w betrachten lassen. Diese Betrachtung ermöglicht viele Aussagen über das Verhalten von Rankern auf Wörtern der entsprechenden Form. Einige solcher Aussagen fassen die Lemmata in diesem Unterabschnitt zusammen. Zunächst ist es jedoch sinnvoll folgenden Konvention zu vereinbaren:

Konvention: Ist w ein π-Term und μ ein Ordnungstyp, dann bezeichnet w_μ das Wort $[\![w]\!]_{\mu+\mu^*}$. Sofern aus dem Kontext zu erkennen ist, welcher π-Term gemeint ist, wird außerdem ${}^{\mu}r$ statt ${}^{w_\mu}r$ für einen Ranker r und ${}^{\mu}r^{\mathrm{c}}$ statt ${}^{w_\mu}r^{\mathrm{c}}$ für einen kondensierten Ranker r^{c} geschrieben.

Es sei hervorgehoben, dass w_μ gleich $[\![w]\!]_{\mu+\mu^*}$ ist und nicht etwa gleich $[\![w]\!]_\mu$. Dies mag zunächst verwirrend wirken, im Folgenden ist aber stets diese Form von Bedeutung. Die Schreibweise lässt sich daher auf diese Weise deutlich verkürzen.

Nun aber zu den Aussagen über Ranker und π-Terme:

Lemma 9: Sei w ein π-Term und sei $\mu \in \mathbb{N} \cup \{\omega\}$ ein Ordnungstyp.
Es gilt für alle a aus einem Alphabet Σ:

$$\begin{aligned}
{}^{\mu}X_a(\pm\infty) = p \neq \bot \implies & \text{ Für alle } \pi\text{-Knoten } v \text{, an denen } p \text{ definiert ist, gilt:} \\
& \; p(v) = 1. \\
{}^{\mu}Y_a(\pm\infty) = p \neq \bot \implies & \text{ Für alle } \pi\text{-Knoten } v \text{, an denen } p \text{ definiert ist, gilt:} \\
& \; p(v) = -1.
\end{aligned}$$

Sei $p \neq \pm\infty$ ein $(\mu + \mu^*)$-Pfad in $G(w)$ und sei $\hat{m} := \max\{p(v) : p(v) \in \mathbb{N}\}$ und $\check{m} := \min\{p(v) : p(v) \in -\mathbb{N}\}$.

Ist $\mu = \omega$ oder $\mathbb{N} \ni \mu > \hat{m}$, so gilt:

$$\begin{aligned}{}^{\mu}X_a(p) = p' \neq \bot \implies & \text{Für alle } \pi\text{-Knoten } v, \text{ an denen } p \text{ definiert ist, gilt:} \\ & p'(v) = 1,\ p'(v) = p(v) \text{ oder } p'(v) = p(v) + 1.\end{aligned}$$

Ist $\mu = \omega$ oder $\mathbb{N} \ni \mu > -\check{m}$, so gilt:

$$\begin{aligned}{}^{\mu}Y_a(p) = p' \neq \bot \implies & \text{Für alle } \pi\text{-Knoten } v, \text{ an denen } p \text{ definiert ist, gilt:} \\ & p'(v) = -1,\ p'(v) = p(v) \text{ oder } p'(v) = p(v) - 1.\end{aligned}$$

Beweis. Sei zuerst ${}^{\mu}X_a(\pm\infty) = p \neq \bot$. Sei v ein π-Knoten in $G(w)$, sodass $p(v) \neq \bot$. Angenommen es wäre $p(v) \neq 1$, also $p(v) > 1$ oder $p(v) \in -\mathbb{N}$. Sei $\tilde{p}$ gleich zu p, aber mit $\tilde{p}(v) := 1$. Dann ist $\tilde{p} <_{\mu+\mu^*,w} p$, was einen Widerspruch zur Definition von ${}^{\mu}X_a$ über das Minimum darstellt.

Sei nun $\mu = \omega$ oder $\mu > \hat{m}$ und sei ${}^{\mu}X_a(p) = p' \neq \bot$. Sei v_p der erste Knoten auf p, an dem p und p' nicht mehr übereinstimmen, und sei $v_{p'}$ der entsprechende Knoten auf p'. Für alle π-Knoten v davor, gilt dann $p(v) = p'(v)$. Gilt $v_p = v_{p'}$, so ist $v_{p'}$ ein π-Knoten. Man beachte, dass für alle π-Knoten v, über die p verläuft, $p(v) + 1 \in S + -S$ für $\mu = \operatorname{ord}(S, \leq_S)$ mit $S \subseteq \mathbb{N}$ ist. Angenommen es ist $p'(v_p) \neq p(v_p) + 1$. Da nach Definition der Ranker $p' >_{\mu+\mu^*,w} p$ gelten muss, ist dann $p'(v_p) > p(v_p) + 1$ oder $p'(v_p) \in -\mathbb{N}$. Definiert man den Pfad $\tilde{p}'$ gleich zu p' aber mit $\tilde{p}'(v_p) := p(v_p) + 1$, so ist $p <_{\mu+\mu^*,w} \tilde{p}' <_{\mu+\mu^*,w} p'$. Dies stellt einen Widerspruch zur Definition von ${}^{\mu}X_a$ über das Minimum dar. Ist v ein π-Knoten auf p' nach $v_{p'}$ oder gilt $v_p \neq v_{p'}$ und $v_{p'} = v$, so ist $p'(v) = 1$, da sonst $\tilde{p}'$ mit $\tilde{p}'(v) := 1$ aber sonst gleich zu p' wieder einen Pfad größer als p, aber kleiner als p' und damit einen Widerspruch liefert.

Die Aussagen zu Y_a folgen mit den entsprechenden links-rechts-dualen Beweisen. □

Man beachte, dass er letzte Beweis tatsächlich eine genauere Aussage liefert als das Lemma sie formuliert: Ob $p'(v) = 1$, $p'(v) = p(v)$ oder $p'(v) = p(v) + 1$ gilt, hängt nämlich von der Position von v auf dem Pfad im Verhältnis zu $v_{p'}$ ab. Dies gilt natürlich auch für die entsprechende Aussage zu ${}^{\mu}Y_a$.

Die erste Aussage führt direkt auf die zweite, die einen Zusammenhang zwischen Rankerlänge und den Werten an π-Knoten eines $(\omega + \omega^*)$-Pfades herstellt:

Lemma 10: Sei w ein π-Term, $\mu \in \mathbb{N} \cup \{\omega\}$ ein Ordnungstyp und $r = Z_1 Z_2 \dots Z_n$ ein Ranker der Länge n.
Ist ${}^{\mu}r(\pm\infty) = p \neq \bot$, so gilt für alle π-Knoten v aus $G(w)$, über die p verläuft:

$$\begin{aligned} p(v) \in \mathbb{N} &\implies p(v) \leq n \text{ und} \\ p(v) \in -\mathbb{N} &\implies p(v) \geq -n. \end{aligned}$$

Beweis. Die Aussage ergibt sich aus Lemma 9. □

Die letzte Aussage dieses Unterabschnitts beschäftigt sich nun mit dem Übergang von ω auf eine (endliche) natürliche Zahl:

Lemma 11: Sei w ein π-Term und Σ ein Alphabet.
Für einen Ranker r der Länge n über Σ gilt:

$$\forall k \geq n : {}^{\omega}r^{\mathrm{c}}(\pm\infty) = {}^{k}r^{\mathrm{c}}(\pm\infty),$$

insbesondere ist

$$\forall k \geq n : r \models^{\mathrm{c}} w_\omega \iff r \models^{\mathrm{c}} w_k.$$

Beweis. Der Beweis erfolgt durch Induktion über die Rankerlänge n. Sei $n = 1$ und $r = Z_a$ mit $Z \in \{X, Y\}$. Es gilt dann für $k \geq 1$:

$$\begin{aligned} Z_a \not\models^{\mathrm{c}} w_\omega &\iff Z_a \not\models w_\omega \\ &\iff a \notin \alpha(w_\omega) = \alpha(w_k) \\ &\iff Z_a \not\models w_k \iff Z_a \not\models^{\mathrm{c}} w_k \end{aligned}$$

Ist ${}^{\omega}Z_a{}^{\mathrm{c}}(\pm\infty) = (-\infty, p_\omega, \infty) \neq \bot$ definiert, so ist also auch ${}^{k}Z_a{}^{\mathrm{c}}(\pm\infty) = (-\infty, p_k, +\infty)$ $\neq \bot$ definiert. Nach Lemma 9 gilt für einen π-Knoten v aus $G(w)$, über den p_ω verläuft, $p_\omega(v) \in \{1, -1\}$. Damit ist p_ω nicht nur ein $(\omega + \omega^*)$-Pfad in $G(w)$, sondern auch ein $(k + k^*)$-Pfad. Angenommen es gilt $p_\omega <_{k+k^*,w} p_k$ oder $p_k <_{k+k^*,w} p_\omega$, also $p_\omega \neq p_k$, so ist dies eine Widerspruch zur Minimalität (falls $Z_a = X_a$) bzw. zur Maximalität (falls $Z_a = Y_a$) der Pfade.

Vor dem Induktionsschritt soll zunächst eine Aussage als Hilfsmittel bewiesen werden: Sind $\check{q}$ und $\hat{q}$ zwei $(k + k^*)$-Pfade in $G(w)$ mit $k \in \mathbb{N}$, sind sie natürlich auch $(\omega + \omega^*)$-Pfade in $G(w)$. Es soll gezeigt werden, dass sich auch die Anordnung überträgt, d. h. dass

$$\check{q} <_{k+k^*,w} \hat{q} \implies \check{q} <_{\omega+\omega^*,w} \hat{q}$$

gilt. Sei dazu $\check{v}$ der erste Knoten auf $\check{q}$, an dem sich $\check{q}$ und $\hat{q}$ unterscheiden; sei $\hat{v}$ der entsprechende Knoten auf $\hat{q}$. Ist $\check{v} \neq \hat{v}$, dann ist $\check{v} < \hat{v}$ entsprechend der Knotenordnung in $G(w)$, falls $\check{q} <_{k+k^*,w} \hat{q}$ ist. Dann gilt aber auch $\check{q} <_{\omega+\omega^*,w} \hat{q}$. Ist $\check{v} = \hat{v}$, so handelt es sich um einen π-Knoten. Sei $k + k^* = \mathrm{ord}\,(\{-k, -k+1, \ldots, -1, 1, 2, \ldots, k\}, \leq_{k+k^*})$ und $\omega + \omega^* = \mathrm{ord}\,(\mathbb{N} + -\mathbb{N}, \leq_{\omega+\omega^*})$. Es ist klar, dass $i \leq_{k+k^*} j \implies i \leq_{\omega+\omega^*} j$ gilt. Ist also $\check{q} <_{k+k^*,w} \hat{q}$, so ist $\check{q}(\check{v}) \leq_{k+k^*} \hat{q}(\check{v})$ und damit $\check{q}(\check{v}) \leq_{\omega+\omega^*} \hat{q}(\check{v})$ und schließlich $\check{q} <_{\omega+\omega^*} \hat{q}$.

Sei die Aussage also nun für alle Ranker der Länge $< n$ gezeigt. Sei $r = r'Z_a$ für $Z \in \{X, Y\}$ und damit ${}^{\omega}r'^{\mathrm{c}}(\pm\infty) = {}^{k}r'^{\mathrm{c}}(\pm\infty) = p_\bot$ für $k \geq n$. Ist $p_\bot = \bot$, so gilt auch ${}^{\omega}r^{\mathrm{c}}(\pm\infty) = {}^{k}r^{\mathrm{c}}(\pm\infty) = \bot$. Ist $p_\bot = (l, p, r) \neq \bot$, betrachte ${}^{k}r^{\mathrm{c}}(\pm\infty) = {}^{k}Z_a{}^{\mathrm{c}}(l, p, r) = p_{k,\bot}$ und ${}^{\omega}r^{\mathrm{c}}(\pm\infty) = {}^{\omega}Z_a{}^{\mathrm{c}}(l, p, r) = p_{\omega,\bot}$. Ist $p_{k,\bot} = p_{\omega,\bot} = \bot$, so ist nichts mehr zu zeigen. Ist $p_{k,\bot} = (l_k, p_k, r_k) \neq \bot$, so gilt es die möglichen Fälle für Z_a zu unterscheiden. Ist $Z_a = X_a$, so ist $l_k = p$ und $r_k = r$. Beides sind $(\omega + \omega^*)$-Pfade in $G(w)$ (tatsächlich kann auch $r = +\infty$ gelten). Außerdem ist offensichtlich p_k ebenfalls ein $(\omega+\omega^*)$-Pfad in $G(w)$. Damit ist ${}^{\omega}X_a(p) = \min_{\leq_{\omega+\omega^*,w}}\{q \in P_{\omega+\omega^*}(w) : w_\omega(q) = a \text{ und } p <_{\omega+\omega^*,w} q\} =:$ $p' \neq \bot$ definiert, da sich $l_k = p <_{k+k^*,w} p_k$ nach der Aussage oben zu $l_k = p <_{\omega+\omega^*,w} p_k$ überträgt und damit $p_k \in \{p' \in P_{\omega+\omega^*}(w) : w_\omega(p') = a \text{ und } p <_{\omega+\omega^*,w} p'\} \neq \emptyset$ ist. Ebenso überträgt sich die Zulässigkeit, es gilt: $l_k = p <_{\omega+\omega^*,w} p' \leq_{\omega+\omega^*,w} p_k <^{+\infty}_{\omega+\omega^*,w}$

$r_k = r$. Für $Z_a = Y_a$ gilt der duale Beweis. Ist also $p_{k,\perp} = (l_k, p_k, r_k) \neq \perp$, so ist $p_{\omega,\perp} = (l_\omega, p_\omega, r_\omega) \neq \perp$. Nach Ranker-Definition gilt $l_k = p = l_\omega$ und $r_k = r = r_\omega$. Außerdem ist p_ω ein $(k + k^*)$-Pfad in $G(w)$ nach Lemma 10. Nimmt man nun an, dass $p_k \neq p_\omega$ gilt, so ist einer der beiden Pfade bezüglich $\leq_{\omega+\omega^*,w}$ echt kleiner, was einen Widerspruch zur Minimalität bzw. zur Maximalität darstellt. Es verbleibt noch die letzte Möglichkeit $p_{\omega,\perp} = (l_\omega, p_\omega, r_\omega) \neq \perp$, aber $p_{k,\perp} = \perp$. Hier lässt sich aber die selbe Argumentation anwenden: p_ω ist wieder nach Lemma 10 ein $(k + k^*)$-Pfad und die Anordnungen übertragen sich, also muss $p_{k,\perp} \neq \perp$ sein und die Werte bereits gleich. $\square$

2.1.2 Ranker und Gleichungen

Dieser Unterabschnitt wird nun zeigen, dass sich die Fragestellung des Wortproblems für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie sowie von $\mathbf{DA}$ durch die Frage nach der Unterscheidbarkeit zweier π-Terme durch bestimmte kondensierte Ranker entscheiden lässt.

Lemma: Seien u und v zwei π-Terme über dem selben Alphabet Σ. Es gilt für $m \geq 1$:

$$\begin{aligned}
[\![u]\!]_{\omega+\omega^*} \equiv_m^X [\![v]\!]_{\omega+\omega^*} &\implies u = v \text{ gilt in } \boldsymbol{R_m} \\
[\![u]\!]_{\omega+\omega^*} \equiv_m^Y [\![v]\!]_{\omega+\omega^*} &\implies u = v \text{ gilt in } \boldsymbol{L_m} \\
[\![u]\!]_{\omega+\omega^*} \equiv_m [\![v]\!]_{\omega+\omega^*} &\implies u = v \text{ gilt in } \boldsymbol{R_m} \vee \boldsymbol{L_m}
\end{aligned}$$

Beweis. Der Beweis ist strukturell für alle drei Aussagen äquivalent, er wird daher hier nur für $\boldsymbol{R_m}$ ausgeführt.

Es gelte $[\![u]\!]_{\omega+\omega^*} \equiv_m^X [\![v]\!]_{\omega+\omega^*}$ für ein $m \geq 1$. Sei $M \in \boldsymbol{R_m}$ ein endliches Monoid. Es ist zu zeigen, dass $u = v$ in M gilt, d. h. dass $\sigma([\![u]\!]_{M!}) = \sigma([\![v]\!]_{M!})$ für jede Belegung $\sigma : \Sigma \to M$ gilt. Nach Theorem 1 gibt es eine natürliche Zahl n (in Abhängigkeit von M), sodass für alle endlichen Wörter $w, w' \in \Sigma^*$ gilt: $w \equiv_{m,n}^X w' \implies \sigma(w) = \sigma(w')$. (Für $\boldsymbol{R_m}$ gilt $n := (m + 1)|M| - m$.) Wähle nun $c \in \mathbb{N}$ so, dass $c(M!) \geq n$ gilt. Nach Lemma 2 ist $(c(M!))^* = c(M!)$ und zusammen mit Lemma 3 gilt dann $(cM!) + (cM!)^* = 2c(M!)$, was insbesondere endlich ist. Sei $r \in R_{m,n}^X \subseteq R_m^X$ ein beliebiger Ranker, dann gilt $r \models^{\mathrm{c}} [\![u]\!]_{\omega+\omega^*} \iff r \models^{\mathrm{c}} [\![v]\!]_{\omega+\omega^*}$ und außerdem:

$$\begin{aligned}
r \models^{\mathrm{c}} [\![u]\!]_{cM!+(cM!)^*} &\iff r \models^{\mathrm{c}} [\![u]\!]_{\omega+\omega^*} && \text{(Lemma 11)} \\
&\iff r \models^{\mathrm{c}} [\![v]\!]_{\omega+\omega^*} \\
&\iff r \models^{\mathrm{c}} [\![v]\!]_{cM!+(cM!)^*} && \text{(Lemma 11)}
\end{aligned}$$

Es gilt also für die beiden (endlichen) Wörter:

$$[\![u]\!]_{2cM!} = [\![u]\!]_{cM!+(cM!)^*} \equiv_{m,n}^X [\![v]\!]_{cM!+(cM!)^*} = [\![v]\!]_{2cM!} \in \Sigma^*$$

Also gilt:

$$\begin{aligned}
\sigma([\![u]\!]_{M!}) &= \sigma([\![u]\!]_{2cM!}) && \text{(Lemma 7)} \\
&= \sigma([\![v]\!]_{2cM!}) && \text{(Theorem 1)} \\
&= \sigma([\![v]\!]_{M!}) && \text{(Lemma 7)}
\end{aligned}$$

$\square$

Lemma: Seien u und v zwei π-Terme über dem selben Alphabet Σ. Es gilt für $m \geq 1$:

$$\begin{aligned}
[\![u]\!]_{\omega+\omega^*} \not\equiv_m^X [\![v]\!]_{\omega+\omega^*} &\implies u = v \text{ gilt in } \boldsymbol{R_m} \text{ nicht} \\
[\![u]\!]_{\omega+\omega^*} \not\equiv_m^Y [\![v]\!]_{\omega+\omega^*} &\implies u = v \text{ gilt in } \boldsymbol{L_m} \text{ nicht} \\
[\![u]\!]_{\omega+\omega^*} \not\equiv_m [\![v]\!]_{\omega+\omega^*} &\implies u = v \text{ gilt in } \boldsymbol{R_m} \vee \boldsymbol{L_m} \text{ nicht}
\end{aligned}$$

Beweis. Wieder wird der Beweis nur exemplarisch für $\boldsymbol{R_m}$ geführt.
Es gelte $[\![u]\!]_{\omega+\omega^*} \not\equiv_m^X [\![v]\!]_{\omega+\omega^*}$, dann gibt es einen Ranker $r \in R_m^X$ mit (ohne Einschränkung) $r \models^c [\![u]\!]_{\omega+\omega^*}$, aber $r \not\models^c [\![v]\!]_{\omega+\omega^*}$. Sei n die Länge dieses Rankers, damit ist $r \in R_{m,n}^X$. Nach Lemma 11 gilt dann

$$\forall k \geq n : r \models^c [\![u]\!]_{k+k^*} \text{ und } r \not\models^c [\![v]\!]_{k+k^*}$$

und somit

$$\forall k \geq n : [\![u]\!]_{k+k^*} \not\equiv_{m,n}^X [\![v]\!]_{k+k^*}.$$

Definiere $M := \Sigma^*/{\equiv_{m,n}^X}$. Nach Theorem 1 ist $M \in \boldsymbol{R_m}$. Angenommen es gilt nun $u = v$ in $\boldsymbol{R_m}$, dann gilt $u = v$ insbesondere in M. Dann gilt also $\sigma([\![u]\!]_{M!}) = \sigma([\![v]\!]_{M!})$ für jede Belegung $\sigma : \Sigma \to M$. Wähle $c \in \mathbb{N}$ so, dass $cM! \geq n$ ist. Es gilt dann:

$$\begin{aligned}
\sigma([\![u]\!]_{cM!+(cM!)^*}) &= \sigma([\![u]\!]_{2cM!}) = \sigma([\![u]\!]_{M!}) && \text{(Lemma 2, Lemma 3), (Lemma 7)} \\
&= \sigma([\![v]\!]_{M!}) = \sigma([\![v]\!]_{2cM!}) && \text{(Lemma 7)} \\
&= \sigma([\![v]\!]_{cM!+(cM!)^*}) && \text{(Lemma 2, Lemma 3)}
\end{aligned}$$

Wähle σ als die natürliche Projektion von Σ nach M, womit mit der Aussage von eben

$$[\![u]\!]_{cM!+(cM!)^*} \equiv_{m,n}^X [\![v]\!]_{cM!+(cM!)^*}$$

gilt. Dies ist ein Widerspruch, da $cM! \geq n$ ist! $\square$

Die beiden letzten Lemmata liefern zusammen also die gesuchte Aussage: Um zu entscheiden, ob eine Gleichung $u = v$ über $\boldsymbol{R_m}$, $\boldsymbol{L_m}$ oder $\boldsymbol{R_m} \vee \boldsymbol{L_m}$ gilt, reicht es zu entscheiden, ob sich $[\![u]\!]_{\omega+\omega^*}$ und $[\![v]\!]_{\omega+\omega^*}$ durch gewisse kondensierte Ranker unterscheiden lassen. Zusammen mit einer kleinen Erweiterung ist dies der Inhalt des folgenden Theorems:

Theorem 3: Seien u und v zwei π-Terme über dem selben Alphabet Σ. Es gilt für $m \geq 1$:

$$\begin{aligned}
[\![u]\!]_{\omega+\omega^*} \equiv_m^X [\![v]\!]_{\omega+\omega^*} &\iff u = v \text{ gilt in } \boldsymbol{R_m} \\
[\![u]\!]_{\omega+\omega^*} \equiv_m^Y [\![v]\!]_{\omega+\omega^*} &\iff u = v \text{ gilt in } \boldsymbol{L_m} \\
[\![u]\!]_{\omega+\omega^*} \equiv_m [\![v]\!]_{\omega+\omega^*} &\iff u = v \text{ gilt in } \boldsymbol{R_m} \vee \boldsymbol{L_m}
\end{aligned}$$

Außerdem gilt:

$$(\forall m \in \mathbb{N} : [\![u]\!]_{\omega+\omega^*} \equiv_m [\![v]\!]_{\omega+\omega^*}) \iff u = v \text{ gilt in } \mathbf{DA}$$

Beweis. Der erste Teil entspricht den Aussagen der beiden Lemmata. Der zweite Teil ergibt sich zusammen mit Theorem 2 aus dem ersten. $\square$

2.2 Normalisierung

Das Wort $[\![w]\!]_{\omega+\omega^*}$ zu einem π-Term w ist im Allgemeinen nicht endlich. Dass sich die Frage nach der Gültigkeit einer Gleichung $u = v$ in den Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie auf die Unterscheidbarkeit durch gewisse kondensierte Ranker zurückführen lässt, scheint daher zunächst keine Verbesserung darzustellen. Tatsächlich ist aber nur eine Betrachtung endlicher Wörter nötig. Diese Einsicht basiert im Wesentlichen darauf, dass sich ein Tripel (l, p, r) von $(\omega + \omega^*)$-Pfaden in einem gewissen Sinne normalisieren lässt. Mit dieser Normalisierung beschäftigt sich nun dieser Abschnitt.

2.2.1 Definition

Die Normalisierung eines Tripels (l, p, r) von $(\omega+\omega^*)$-Pfaden wird den Verlauf der Pfade nicht ändern, wohl aber die Werte an den π-Knoten. Ziel ist es dabei die Abstände zwischen den drei Pfaden an jedem Knoten zu erhalten. Wie genau dies vonstattengeht, klärt dieser Unterabschnitt.

Tatsächlich lassen sich nicht alle zulässigen Tripel aus $(\omega + \omega^*)$-Pfaden in der Form, wie sie hier besprochen wird, normalisieren. Um jene, die sich normalisieren lassen, beschreiben zu können ist die folgende Definition notwendig:

Definition 1: Sei w ein π-Term über einem Alphabet Σ. Definiere $\bar{P}_{\omega+\omega^*}(w)$ als die Menge aller zulässigen $(l, p, r) \in (P_{\omega+\omega^*}(w) + \{-\infty\}) \times P_{\omega+\omega^*}(w) \times (P_{\omega+\omega^*}(w) + \{+\infty\})$, die die folgenden beiden Eigenschaften erfüllen:

$$\begin{array}{lll}
\text{I.} & p(v) \in \mathbb{N},\quad p(v) > 1 & \Longrightarrow l(v) \in \mathbb{N}^{\dagger},\ 0 \leq p(v) - l(v) \leq 1 \\
 & & \text{und vor } v \text{ stimmen } p \text{ und } l \text{ überein} \\
 & p(v) \in -\mathbb{N},\ p(v) < -1 & \Longrightarrow r(v) \in -\mathbb{N},\ 0 \leq r(v) - p(v) \leq 1 \\
 & & \text{und vor } v \text{ stimmen } p \text{ und } r \text{ überein} \\
\text{II.} & r(v) \in \mathbb{N},\quad r(v) > 1 & \Longrightarrow l(v), p(v) \in \mathbb{N},\ 0 \leq r(v) - l(v) \leq 1 \\
 & & \text{und vor } v \text{ stimmt } r \text{ mit } l \text{ und } p \text{ überein} \\
 & l(v) \in -\mathbb{N},\ l(v) < -1 & \Longrightarrow p(v), r(v) \in -\mathbb{N},\ 0 \leq r(v) - l(v) \leq 1 \\
 & & \text{und vor } v \text{ stimmt } l \text{ mit } p \text{ und } r \text{ überein}
\end{array}$$

Dabei soll $-\infty(v)$ und $+\infty(v)$ für alle $v \in G(w)$ undefiniert sein.
Ferner sei $\bar{P}_{\omega+\omega^*}$ die Vereinigung der Mengen $\bar{P}_{\omega+\omega^*}(w)$ für alle π-Terme w über beliebigem Alphabet.

An die Elemente (l, p, r) in $\bar{P}_{\omega+\omega^*}$ wird also im Wesentlichen die Forderung gestellt, dass die Pfade – sofern gewisse lokale Voraussetzungen erfüllt sind – über die selben π-Knoten verlaufen und sich die Werte dort höchstens um 1 unterscheiden. Es wird sich später herausstellen, dass jene Tripel, die durch Anwendung eines kondensierten Rankers entstehen, die geforderten Eigenschaften erfüllen.

Mit der Definition von $\bar{P}_{\omega+\omega^*}$ lässt sich nun auch die Normalisierung definieren:

†Insbesondere ist $l(v) \neq \perp$ und l verläuft über v.

Definition (Normalisierung)**:** Sei $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$ für einen π-Term w über dem Alphabet Σ. Definiere $\overline{(l, p, r)} = (\bar{l}, \bar{p}, \bar{r})$ folgendermaßen: Ist $l = -\infty$, so ist auch $\bar{l} := -\infty$. Ist $r = +\infty$, so ist auch $\bar{r} := +\infty$. Ansonsten definiere:

$$\begin{aligned} \bar{l} : G(w) &\to_p \mathbb{N} + (-\mathbb{N}) + \{\top\} \\ v &\mapsto \begin{cases} \top & \text{falls } l(v) = \top \\ 1 & \text{falls } l(v) \in \mathbb{N} \\ -1 & \text{falls } l(v) = -1 \\ l(v) - r(v) - 1 & \text{falls } l(v) \in -\mathbb{N} \text{ und } l(v) < -1 \end{cases} \end{aligned}$$

$$\begin{aligned} \bar{r} : G(w) &\to_p \mathbb{N} + (-\mathbb{N}) + \{\top\} \\ v &\mapsto \begin{cases} \top & \text{falls } r(v) = \top \\ -1 & \text{falls } r(v) \in -\mathbb{N} \\ 1 & \text{falls } r(v) = 1 \\ r(v) - l(v) + 1 & \text{falls } r(v) \in \mathbb{N} \text{ und } r(v) > 1 \end{cases} \end{aligned}$$

$$\begin{aligned} \bar{p} : G(w) &\to_p \mathbb{N} + (-\mathbb{N}) + \{\top\} \\ v &\mapsto \begin{cases} \top & \text{falls } p(v) = \top \\ 1 & \text{falls } p(v) = 1 \\ -1 & \text{falls } p(v) = -1 \\ p(v) - l(v) + 1 & \text{falls } p(v) \in \mathbb{N} \text{ und } p(v) > 1 \\ p(v) - r(v) - 1 & \text{falls } p(v) \in -\mathbb{N} \text{ und } p(v) < -1 \end{cases} \end{aligned}$$

Die normalisierten Pfade verlaufen also stets über die selben Knoten wie ihre nicht normalisieren Gegenstücke. Lediglich die Werte der Pfade an den π-Knoten werden verändert. Dazu beachte man, dass die zugeordneten Werte nach Definition 1 stets definiert und aus $\{1, 2, -2, -1\}$ sind.

2.2.2 Eigenschaften

Einige interessante Eigenschaften der Normalisierung ergeben sich direkt aus der Definition. Das folgende Lemma, das solche Eigenschaften zusammenfasst, erfolgt daher ohne Beweis:

Lemma: Sei $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$ für einen π-Term w über dem Alphabet Σ, sei $\overline{(l, p, r)} = (\bar{l}, \bar{p}, \bar{r})$ und sei v ein Knoten in $G(w)$.
Es gilt:

- l und $\bar{l}$, p und $\bar{p}$ sowie r und $\bar{r}$ verlaufen jeweils über die selben Knoten.
- $l(v) \in \mathbb{N} \iff \bar{l}(v) \in \mathbb{N} \quad p(v) \in \mathbb{N} \iff \bar{p}(v) \in \mathbb{N} \quad r(v) \in \mathbb{N} \iff \bar{r}(v) \in \mathbb{N}$
  $l(v) \in -\mathbb{N} \iff \bar{l}(v) \in -\mathbb{N} \quad p(v) \in -\mathbb{N} \iff \bar{p}(v) \in -\mathbb{N} \quad r(v) \in -\mathbb{N} \iff \bar{r}(v) \in -\mathbb{N}$
- $l = -\infty \iff \bar{l} = -\infty$ und $r = +\infty \iff \bar{r} = +\infty$
- $\bar{l}$, $\bar{p}$, $\bar{r}$ sind $(2 + 2^*)$-Pfade in $G(w)$. ($\bar{l} = -\infty$ und $\bar{r} = +\infty$ sind hierbei natürlich ausgenommen.)

Während sich diese Eigenschaften direkt ergeben, bedürfen andere einer genaueren Betrachtung. Eine dieser Eigenschaften beschreibt das folgende Lemma:

Lemma 12: Sei $(l, p, r) \in \bar{P}_{\omega+\omega^*}$ und $\overline{(l, p, r)} = (\bar{l}, \bar{p}, \bar{r})$. Es unterscheiden sich l und p sowie p und r jeweils am selben Knoten wie $\bar{l}$ und $\bar{p}$ bzw. $\bar{p}$ und $\bar{r}$ zum ersten Mal, sofern $l \neq -\infty$ bzw. $r \neq +\infty$. Außerdem überträgt sich die Zulässigkeit von (l, p, r) auf $(\bar{l}, \bar{p}, \bar{r})$.

Beweis. Zunächst zu l und p: Sei v_l der erste Knoten auf l, an dem sich l und p unterscheiden. Sei v_p der entsprechende Knoten auf p. Sei v ein π-Knoten oberhalb von v_l bzw. v_p. Gilt dort $p(v) = l(v) \in \mathbb{N}$, so ist nach Definition $\bar{l}(v) = 1$. Ist $p(v) = 1$, so ist außerdem $\bar{p}(v) = 1$. Ist $p(v) > 1$, so ist $\bar{p}(v) = p(v) - l(v) + 1 = 1$. In beiden Fällen gilt also $\bar{p}(v) = \bar{l}(v)$. Gilt an v stattdessen $p(v) = l(v) \in -\mathbb{N}$, so muss unterschieden werden: Ist $p(v) = l(v) = -1$, dann ist nach Definition $\bar{l}(v) = -1 = \bar{p}(v)$. Ist $p(v) = l(v) < -1$, so ist $\bar{l}(v) = l(v) - r(v) - 1 = p(v) - r(v) - 1 = \bar{p}(v)$. l und p stimmen also oberhalb von v_l bzw. v_p überein.

Ist nun $v_l \neq v_p$, so ist nichts mehr zu zeigen. Ist $v_l = v_p$ ein π-Knoten, so muss aufgrund der Zulässigkeit von (l, p, r) gelten: $l(v_l) <_{\omega+\omega^*} p(v_p)$. Dafür gibt es folgende Möglichkeiten:

- $l(v_l), p(v_p) \in \mathbb{N}$: Dann ist $\bar{l}(v_l) = 1$ und $\bar{p}(v_p) = p(v_p) - l(v_l) + 1 > 1$, da $p(v_p) > l(v_l) \geq 1$ sein muss.
- $l(v_l) \in \mathbb{N}, p(v_p) \in -\mathbb{N}$: Nach Definition ist $\bar{l}(v_l) = 1$ und $\bar{p}(v_p) \in -\mathbb{N}$.
- $l(v_l), p(v_p) \in -\mathbb{N}$: Es muss $l(v_l) \neq -1$ bzw. $l(v_l) < -1$ gelten. Ist $p(v_p) = -1$, so ist $\bar{p}(v_p) = -1$ und nach Definition 1 $r(v) = -1$. Also ist $\bar{l}(v_l) = l(v) - r(v) - 1 = l(v) < -1$. Ist $p(v_p) < -1$, so ist $\bar{p}(v_p) = p(v_p) - r(v_p) - 1 > l(v_l) - r(v_l) - 1 = \bar{l}(v_l)$.

In allen Fällen unterscheiden sich $\bar{l}$ und $\bar{p}$ in ihren Werten an $v_l = v_p$, es gilt jeweils sogar $\bar{l}(v_l) <_{\omega+\omega^*} \bar{p}(v_p)$. Damit ist v_l auch der erste Knoten auf $\bar{l}$, an dem sich $\bar{l}$ und $\bar{p}$ unterscheiden. Entsprechendes gilt auch für v_p. Schließlich gilt also $\bar{l} <_{2+2^*,w} \bar{p}$.

Die Aussage für p und r verhält sich dual. Zusammen ergibt sich damit die Zulässigkeit von $(\bar{l}, \bar{p}, \bar{r})$. □

Als nächstes soll die Menge $\bar{P}_{\omega+\omega^*}$ genauer untersucht werden. Eine der wichtigsten Aussagen dabei liefert das folgende Lemma:

Lemma: Sei w ein π-Term über dem Alphabet Σ und sei Z ein Ranker der Länge 1. Ist ${}^{\omega}Z^{\mathrm{c}}(\pm\infty) = (l, p, r) \neq \perp$, so ist $(l, p, r) \in \bar{P}_{\omega+\omega^*}$. Außerdem ist für alle $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$ auch ${}^{\omega}Z^{\mathrm{c}}(l, p, r)$ in $\bar{P}_{\omega+\omega^*}$, sofern dies definiert ist.

Beweis. Zunächst zum ersten Teil der Aussage. Ist $Z = X_a$ mit $a \in \Sigma$, so ist aufgrund der Definition über das Minimum $p(v) = 1$, falls $p(v)$ für einen π-Knoten $v \in G(w)$ definiert ist. Für $Z = Y_a$ mit $a \in \Sigma$ ist entsprechend $p(v) = -1$, falls $p(v)$ definiert ist. In beiden Fällen ist I. erfüllt. Außerdem ist für einen kondensierten Ranker der Länge 1 $l = -\infty$ und $r = +\infty$ und damit auch II. erfüllt.

Sei nun $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$. Der Beweis für $Z \in Y_\Sigma$ ist dual zu dem für $Z \in X_\Sigma$, sei daher $Z = X_a$ für ein $a \in \Sigma$. Sei zudem ${}^{\omega}X_a{}^{\mathrm{c}}(l, p, r) = (l', p', r') \neq \perp$. Nach Definition der kondensierten Ranker ist $l' = p$ und $r' = r$. Sei v_p der erste Knoten auf p, an dem p und p' nicht mehr übereinstimmen. Sei $v_{p'}$ der entsprechende Knoten auf p'. Sei $v \in G(w)$

ein beliebiger π-Knoten aus dem gemeinsamen Anfang von p und p'. Nach Wahl von v stimmen p' und $l' = p$ also vor v überein. Ist $p'(v) \in \mathbb{N}$ und $p'(v) > 1$, so gilt: Zunächst ist $l'(v) = p(v) = p'(v) \in \mathbb{N}$, insgesamt gilt:

$$0 \leq p'(v) - l'(v) = p'(v) - p(v) = 0 \leq 1.$$

Ist $p'(v) \in -\mathbb{N}$ und $p'(v) < -1$, so gilt weil $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$

$$0 \leq r'(v) - p'(v) = r(v) - p(v) \leq 1.$$

Damit ist I. für alle Knoten auf dem gemeinsamen Anfang von p und p' gezeigt. Sei v nun ein π-Knoten auf p' (echt) unterhalb von $v_{p'}$. Dort gilt $p'(v) = 1$, da sich sonst aus p' ein bezüglich $\leq_{\omega+\omega^*,w}$ echt kleinerer Pfad $\tilde{p}$ über $\tilde{p}(v) := 1$ konstruieren ließe, was einen Widerspruch zur Minimalität von p' darstellt.

Für I. verbleibt nun nur noch zu zeigen, dass die Aussage auch an $v_{p'}$ gilt. Nach Wahl von $v_{p'}$ stimmen p' und $l' = p$ vor $v_{p'}$ überein. Gilt $v_p \neq v_{p'}$ und ist $v_{p'}$ somit Kind eines Konkatenationsknotens, so ist auch hier $p'(v_{p'}) = 1$, falls es sich um einen π-Knoten handelt, und somit nichts zu zeigen. Ist $v_p = v_{p'}$ und damit sicher ein π-Knoten, gilt es die folgenden Fälle zu unterscheiden:

1. $p(v_p) \in \mathbb{N}$: Aufgrund der Minimalität von p' ist $p'(v_p) = p(v_p) + 1 \in \mathbb{N}$. Es gilt $l'(v_p) = p(v_p) \in \mathbb{N}$ und $0 \leq p'(v_p) - l'(v_p) = p(v_p) + 1 - p(v_p) = 1 \leq 1$.
2. $p(v_p) \in -\mathbb{N}$: Da $p'(v_p) > p(v_p)$ gelten muss, ist $p(v_p) < -1$. Außerdem ist wieder $p'(v_p) = p(v_p) + 1$. Ist $p'(v_p) = -1$, so ist nichts zu zeigen. Ist $p'(v_p) < -1$, so gilt $r'(v_p) = r(v_p) \in -\mathbb{N}$ und $0 \leq r'(v_p) - p'(v_p) = r(v_p) - p(v_p) - 1 \leq 1 - 1 = 0 \leq 1$, wegen $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$.

Um II. zu zeigen, sei nun $v \in G(w)$ ein beliebiger π-Knoten. Gilt $r'(v) = r(v) \in \mathbb{N}$ und $r'(v) = r(v) > 1$, so stimmt $r = r'$ vor v mit l und $p = l'$ überein; dies gilt wegen $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$. Aufgrund der Zulässigkeit von (l', p', r') muss r' dort auch mit p' übereinstimmen. Außerdem ist $l(v) \in \mathbb{N}$, $l'(v) = p(v) \in \mathbb{N}$ und $r(v) - l(v) \leq 1$. Damit ist

$$0 \leq r'(v) - l'(v) = r(v) - \underbrace{p(v)}_{\geq l(v)} \leq r(v) - l(v) \leq 1$$

und somit direkt $p'(v) \in \mathbb{N}$ aufgrund der Zulässigkeit von (l', p', r'). Gilt $l'(v) \in -\mathbb{N}$ und $l'(v) < -1$, so ist wegen $l'(v) = p(v)$ und I. $r'(v) = r(v) \in -\mathbb{N}$ und $0 \leq r'(v) - l'(v) = r(v) - p(v) \leq 1$. Außerdem stimmt $p = l'$ vor v mit $r = r'$ überein. Die Übereinstimmung vor v von l' mit p' und $p'(v) \in -\mathbb{N}$ folgt wieder aus der Zulässigkeit von (l', p', r'). $\square$

Man beachte, dass sich die Aussage des Lemmas folgendermaßen auffassen lässt: Ist r ein Ranker beliebiger Länge und ${}^{\omega}r^{\mathrm{c}}(\pm\infty) = (l, p, r)$ definiert, so ist $(l, p, r) \in P_{\omega+\omega^*}$ und damit $\overline{(l, p, r)}$ definiert.

Zusammen mit dem folgenden Lemma ergibt sich noch eine weitere Aussage darüber, für welche Pfade die Normalisierung definiert ist.

Lemma: Sei $(l, p, r) \in \bar{P}_{\omega+\omega^*}$. Dann ist auch $\overline{(l, p, r)} = (\bar{l}, \bar{p}, \bar{r}) \in \bar{P}_{\omega+\omega^*}$.

Beweis. Nach Lemma 12 ist $(\bar{l}, \bar{p}, \bar{r})$ zulässig. Es ist also nur noch zu zeigen, dass $(\bar{l}, \bar{p}, \bar{r})$ die Eigenschaften I. und II. erfüllt. Ist $\bar{p}(v) \in \mathbb{N}$ und $\bar{p}(v) > 1$ an einem Knoten v, so kann dies nach Definition der Normalisierung nur eintreten, wenn $p(v) - l(v) + 1 > 1$ und $p(v) \in \mathbb{N} \setminus \{1\}$ ist. Dann ist auch $l(v) \in \mathbb{N}$ und damit $\bar{l}(v) = 1 \in \mathbb{N}$. Also ist $0 \leq \bar{p}(v) - \bar{l}(v) = p(v) - l(v) + 1 - 1 = p(v) - l(v) \leq 1$. Da p vor v mit l übereinstimmt und sich $\bar{p}$ und $\bar{l}$ nach Lemma 12 am selben Knoten wie p und l zum ersten Mal unterscheiden, stimmt auch $\bar{p}$ vor v mit $\bar{l}$ überein. Die gleiche Argumentation lässt sich auch für $\bar{p}(v) \in -\mathbb{N}$ und $\bar{p}(v) < -1$ mit für l und r vertauschten Rollen anwenden. Damit ist die Gültigkeit von I. gezeigt.

Gilt an einem Knoten v nun $\bar{r}(v) \in \mathbb{N}$ und $\bar{r}(v) > 1$, so ist dies nur über den Fall $\bar{r} = r(v) - l(v) + 1$ bzw. $r(v) \in \mathbb{N} \setminus \{1\}$ möglich. Dann ist $l(v) \in \mathbb{N}$ und damit $\bar{l}(v) = 1$. Außerdem stimmt r vor v mit l und p überein. Nach Lemma 12 stimmen dort damit auch $\bar{l}$, $\bar{p}$ und $\bar{r}$ überein. Aufgrund der Zulässigkeit von $(\bar{l}, \bar{p}, \bar{r})$ muss dann zudem $\bar{p}(v) \in \mathbb{N}$ gelten. Schließlich gilt auch $0 \leq \bar{r}(v) - \bar{l}(v) = r(v) - l(v) + 1 - 1 \leq 1$. Für $\bar{l}(v) \in -\mathbb{N} \setminus \{-1\}$ gilt wieder der duale Beweis. Damit ist auch II. erfüllt. ☐

Die Normalisierung ist also eine Abbildung von $\bar{P}_{\omega+\omega^*}$ nach $\bar{P}_{\omega+\omega^*}$ und außerdem wird die Menge $\bar{P}_{\omega+\omega^*}$ durch Anwenden eines Rankers (egal ob von der Länge 1 oder länger) nicht verlassen. Es ist also immer möglich das Ergebnis einer Ranker-Anwendung zu normalisieren, sofern die Normalisierung bereits vor Anwendung des Ranker möglich war.

2.2.3 Anwendung

Um die Normalisierung nutzbringend anwenden zu können sind zunächst noch einige weitere Eigenschaften notwendig. Diese dienen dazu von unendlichen Wörtern der Form $[\![w]\!]_{\omega+\omega^*}$ zu einem π-Term w auf endliche Wörter überzugehen.

Lemma: Sei $(l, p, r) \in \bar{P}_{\omega+\omega^*}(w)$ für einen π-Term w über dem Alphabet Σ und sei $\overline{(l, p, r)} = (\bar{l}, \bar{p}, \bar{r})$. Sei außerdem $a \in \Sigma$.
Es gilt:

$$^{[\![w]\!]_{\omega+\omega^*}} X_a{}^{\mathrm{c}}(l, p, r) = (l', p', r') \neq \bot \implies {}^{[\![w]\!]_{3+3^*}} X_a{}^{\mathrm{c}}(\bar{l}, \bar{p}, \bar{r}) = (\bar{l}', \bar{p}', \bar{r}') \neq \bot$$
$$^{[\![w]\!]_{\omega+\omega^*}} Y_a{}^{\mathrm{c}}(l, p, r) = (l', p', r') \neq \bot \implies {}^{[\![w]\!]_{3+3^*}} Y_a{}^{\mathrm{c}}(\bar{l}, \bar{p}, \bar{r}) = (\bar{l}', \bar{p}', \bar{r}') \neq \bot$$

Beweis. Der Beweis der Aussage zu Y_a erfolgt dual zu dem für die Aussage zu X_a. Daher wird hier nur die Aussage zu X_a bewiesen. Dabei ist zunächst zu zeigen, dass ${}^{[\![w]\!]_{3+3^*}} X_a(\bar{p})$ definiert ist.

Betrachte dazu die nicht normalisierten Pfade p und p'. Sei v_p der erste Knoten auf p an dem sich diese Pfade unterscheiden und $v_{p'}$ der entsprechende Knoten auf p'. Ist $v_p = v_{p'}$ und damit ein π-Knoten, so muss $p'(v_p) = p(v_p) + 1$ gelten. Nach Definition der Normalisierung kann $\bar{p}(v_p) = -1$ nicht auftreten: Dazu müsste entweder $p(v_p) = -1$ oder $p(v_p) = r(v_p)$ sein. In beiden Fällen könnte p' bei v_p nicht den Wert $p(v_p) + 1$ annehmen. Definiere nun den Pfad $\tilde{p}$, der über die selben Knoten wie p' verläuft. Vor $v_{p'}$ seien die Werte an den π-Knoten wie bei $\bar{p}$, an $v_{p'} = v_p$ gelte $\tilde{p}(v_p) := \bar{p}(v_p) + 1$

und für einen π-Knoten v echt nach $v_{p'}$ sei $\tilde{p}(v) := 1$. Da $\bar{p}(v_p) \in \{1, 2, -2\}$ gilt, ist $\tilde{p}(v_p) \in \{2, 3, -1\}$. Damit ist $\tilde{p}$ ein $(3 + 3^*)$-Pfad in $G(w)$ und nach Konstruktion gilt $\tilde{p} >_{3+3^*,w} \bar{p}$. Ist $v_p \neq v_{p'}$, so lässt sich ein ähnlicher Pfade definieren: $\tilde{p}$ verläuft dann wieder über die selben Knoten wie p'; vor $v_{p'}$ haben die π-Knoten die selben Werte wie bei $\bar{p}$ und für einen π-Knoten v nach $v_{p'}$ gilt wieder $\tilde{p}(v) := 1$. Auch dann gilt $\tilde{p} >_{3+3^*,w} \bar{p}$. In beiden Fällen ist also die Menge $\{q : [\![w]\!]_{3+3^*}(q) = a \text{ und } \bar{p} <_{3+3^*,w} q\}$ nicht leer und damit ${}^{[\![w]\!]_{3+3^*}}X_a(\bar{p})$ definiert.

Sei also $\bar{p}' := {}^{[\![w]\!]_{3+3^*}}X_a(\bar{p}) \neq \bot$. Nach Definition ist $\bar{p}' >_{3+3^*,w} \bar{p}$. Für die Zulässigkeit von $(\bar{p}, \bar{p}', \bar{r})$ bleibt nur noch zu zeigen, dass $\bar{p}' \leq_{3+3^*,w} \tilde{p} <^{+\infty}_{3+3^*,w} \bar{r}$ gilt, was für $\bar{r} = +\infty$ trivialerweise erfüllt ist. Sei also $r \neq +\infty$ und sei u_p der erste Knoten auf p an dem sich p und r unterscheiden; der entsprechende Knoten auf r sei u_r. Nach Lemma 12 ist u_p auch der erste Knoten auf $\bar{p}$, an dem sich $\bar{p}$ und $\bar{r}$ unterscheiden und u_r auch der entsprechende Knoten auf $\bar{r}$. Liegt u_p (echt) vor v_p auf $\bar{p}$, so verhält sich dort $\tilde{p}$ nach Konstruktion gleich wie $\bar{p}$. Damit gilt aber schon $\tilde{p} <_{3+3^*,w} \bar{r}$. Außerdem kann u_p nicht (echt) nach v_p auf p liegen, da sonst $p' <_{\omega+\omega^*,w} r$ nicht erfüllt sein könnte. Es verbleibt also nur noch den Fall $u_p = v_p$ zu betrachten.

a) Ist $v_p \neq v_{p'}$, so handelt sich dabei um das linke und das rechte Kind eines Konkatenationsknotens. Nach Wahl von $u_p = v_p$ muss p an diesem Konkatenationsknoten noch mit r übereinstimmen, d. h. $u_r = v_{p'}$ muss gelten, da es keine andere Möglichkeit für den weiteren Verlauf von r gibt. Sie nun w' der π-Term, der sich durch den Teilbaum von $G(w)$ mit $v_{p'}$ als Wurzel ergibt. Sei $p'_{w'}$ der Teil von p' der sich durch Einschränkung auf Knoten aus $G(w')$ ergibt; der entsprechende Teil von r sei $r_{w'}$, der von $\bar{r}$ sei $\bar{r}_{w'}$ und der von $\tilde{p}$ sei schließlich $\tilde{p}_{w'}$. Nach Konstruktion von $\tilde{p}$, ist $\tilde{p}_{w'} = p'_{w'}$. Aufgrund der Minimalität von p' muss $p'_{w'} = \min_{\leq_{\omega+\omega^*,w'}}\{q \in P_{\omega+\omega^*}(w') : w'(p) = a\}$ gelten, da sich sonst auch ein bezüglich $\leq_{\omega+\omega^*,w}$ kleinerer Pfad konstruieren ließe. Da p' und r mindestens bis zu $v_{p'}$ übereinstimmen, überträgt sich $p' <_{\omega+\omega^*,w} r$ – was aus der Zulässigkeit von (p, p', r) folgt – auf den Teilbaum $G(w')$. Also gilt $\tilde{p}_{w'} = p'_{w'} <_{\omega+\omega^*,w'} r_{w'}$. Betrachtet man nun die Definition der Normalisierung, so gilt an einem beliebigen π-Knoten v in $G(w)$: $r(v) \in -\mathbb{N} \implies \bar{r}(v) = -1 \geq r(v)$ und $r(v) = 1 \implies \bar{r}(v) = 1 \geq r(v)$. Der Fall $r(v) \in \mathbb{N} \setminus \{1\}$ kann für einen π-Knoten in $G(w')$ nicht auftreten, da sonst nach Definition 1 r und p bis zu v übereinstimmen müssten, was nach Wahl von $u_r = v_{p'}$ als Wurzel von $G(w')$ nicht möglich ist. Also gilt $r_{w'} \leq_{\omega+\omega^*,w'} \bar{r}_{w'}$. Insgesamt ist damit $\tilde{p}_{w'} = p'_{w'} <_{\omega+\omega^*,w'} r_{w'} \leq_{\omega+\omega^*,w'} \bar{r}_{w'}$. Da die Werte von $\tilde{p}_{w'}$ und $\bar{r}_{w'}$ nicht kleiner als -3 oder größer als 3 sein können, gilt also $\tilde{p}_{w'} <_{3+3^*,w'} \bar{r}_{w'}$. Da $\tilde{p}$ und $\bar{r}$ bis zu $u_r = v_{p'}$ gleich sind, überträgt sich die Aussage vom Teilbaum auf ganz $G(w)$, es gilt also $\tilde{p} <_{3+3^*,w} \bar{r}$.

b) Ist $v_p = v_{p'}$ und damit ein π-Knoten, so kann zunächst die Situation auftreten, dass $u_p = v_p = v_{p'}$ das linke und u_r das rechte Kind eines Konkatenationsknotens darstellt. Dann ist aber $\tilde{p} <_{3+3^*,w} \bar{r}$ bereits erfüllt. Ansonsten muss $u_r = u_p = v_p = v_{p'}$ gelten und r verläuft über diesen Knoten. Nach Konstruktion ist $\tilde{p}(v_p) = \bar{p}(v_p) + 1 \leq_{3+3^*} \bar{r}(v_p)$. Falls sogar $\tilde{p}(v_p) <_{3+3^*} \bar{r}(v_p)$ gilt, ist nichts mehr zu zeigen. Ansonsten lässt sich die selbe Argumentation wie eben über den Teilbaum von $G(w)$ mit dem Kind von v_p als Wurzel anwenden.

□

Lemma: Sei $(l,p,r) \in \bar{P}_{\omega+\omega^*}(w)$ für einen π-Term w über dem Alphabet Σ und sei $\overline{(l,p,r)} = (\bar{l},\bar{p},\bar{r})$. Sei außerdem $a \in \Sigma$.
Es gilt:

$$^{[\![w]\!]_{\omega+\omega^*}}X_a{}^{\mathrm{c}}(l,p,r) = (l',p',r') \neq \bot \Longleftarrow {}^{[\![w]\!]_{3+3^*}}X_a{}^{\mathrm{c}}(\bar{l},\bar{p},\bar{r}) = (\bar{l}',\bar{p}',\bar{r}') \neq \bot$$
$$^{[\![w]\!]_{\omega+\omega^*}}Y_a{}^{\mathrm{c}}(l,p,r) = (l',p',r') \neq \bot \Longleftarrow {}^{[\![w]\!]_{3+3^*}}Y_a{}^{\mathrm{c}}(\bar{l},\bar{p},\bar{r}) = (\bar{l}',\bar{p}',\bar{r}') \neq \bot$$

Beweis. Wieder ist die Aussage für Y_a dual zu der für X_a, weshalb auch hier auf einen expliziten Beweis für Y_a verzichtet werden soll.

Damit ${}^{[\![w]\!]_{\omega+\omega^*}}X_a{}^{\mathrm{c}}(l,p,r)$ definiert ist, muss zunächst ${}^{[\![w]\!]_{\omega+\omega^*}}X_a(p)$ definiert sein. Definiere dazu den $(\omega+\omega^*)$-Pfad $\tilde{p}$ wie folgt: Er verläuft über die selben Knoten wie $\bar{p}'$. Sei $v_{\bar{p}}$ der erste Knoten auf $\bar{p}$, an dem sich $\bar{p}$ und $\bar{p}'$ unterscheiden; sei $v_{\bar{p}'}$ der entsprechende Knoten auf $\bar{p}'$. Für einen π-Knoten v auf $\tilde{p}$ (echt) vor $v_{\bar{p}'}$ sei $\tilde{p}(v) := p(v)$. Ist $v_{\bar{p}} = v_{\bar{p}'}$ und damit ein π-Knoten, so definiert $\tilde{p}(v_{\bar{p}'}) := p(v_{\bar{p}'}) + 1$. An einen π-Knoten echt nach $v_{\bar{p}'}$ oder an $v_{\bar{p}'}$ selbst, falls $v_{\bar{p}'} \neq v_{\bar{p}}$ ist, soll $\tilde{p}$ den Wert 1 haben. Nach Definition ist $\tilde{p} >_{\omega+\omega^*,w} p$ und $w(\tilde{p}) = a$. Damit ist ${}^{[\![w]\!]_{\omega+\omega^*}}X_a(p) = p'$ definiert und außerdem $p <_{\omega+\omega^*,w} p'$.

Noch zu zeigen ist: $\tilde{p} <^{+\infty}_{\omega+\omega^*,w} r$. Da nach Definition über das Minimum $p' \leq_{\omega+\omega^*,w} \tilde{p}$ gilt, wäre damit die Zulässigkeit von $(p,p',r) = (l',p',r')$ gezeigt. Für $r = +\infty$ ist nichts mehr zu zeigen, sei also $r \neq +\infty$ und sei u_p der erste Knoten auf p, an dem sich p und r unterscheiden; sei u_r der entsprechende Knoten auf r. Nach Lemma 12 ist u_p auch der erste Knoten auf $\bar{p}$, an dem sich $\bar{p}$ und $\bar{r}$ unterscheiden; Entsprechendes gilt für u_r. Befindet sich u_p echt vor $v_{\bar{p}}$ auf p, so gilt $\tilde{p} <_{\omega+\omega^*,w} r$, weil sich $\tilde{p}$ dort wie p verhält und $p <_{\omega+\omega^*,w} r$ aufgrund der Zulässigkeit von (l,p,r) gilt. Andererseits kann u_p nicht echt nach $v_{\bar{p}}$ auf $\bar{p}$ liegen, da sonst $\bar{p}' >_{3+3^*,w} \bar{r}$ wäre. Es ist nur noch der Fall $u_p = v_{\bar{p}}$ zu betrachten.

a) Gilt $v_{\bar{p}} = v_{\bar{p}'}$ und es handelt sich damit um einen π-Knoten, so muss erneut unterschieden werden: Ist $v_{\bar{p}} = u_p$ das linke Kind und u_r das rechte Kind des selben Konkatenationsknotens, so ist $\tilde{p} <_{\omega+\omega^*,w} r$ erfüllt. Ist $v_{\bar{p}} = v_{\bar{p}'} = u_p = u_r$, so ist $\tilde{p}(v_{\bar{p}'}) = p(v_{\bar{p}}) + 1 \leq_{\omega+\omega^*} r(u_r)$, weil $p(u_p) <_{\omega+\omega^*} r(u_r)$ nach Wahl von u_p bzw. u_r gelten muss. Gilt bereits $\tilde{p}(v_{\bar{p}'}) <_{\omega+\omega^*} r(u_r)$, so ist nicht mehr zu zeigen. Ist $\tilde{p}(v_{\bar{p}'}) = r(u_r)$, so sei w' der π-Term, sodass $G(w')$ dem Unterbaum von $G(w)$ mit dem (einzigen) Kind von $v_{\bar{p}}$ als Wurzel entspricht. Schränkt man $\bar{p}'$ auf Knoten aus $G(w')$ zu $\bar{p}'_{w'}$ ein, so ist $\bar{p}'_{w'}$ wegen der Minimalität gleich der Einschränkung von $\tilde{p}$ auf Knoten aus $G(w')$ zu $\tilde{p}_{w'}$. Für einen π-Knoten v auf dem gemeinsamen Anfang von $\bar{p}'$ und $\bar{r}$ in $G(w')$ gilt daher: $\bar{r}(v) = \bar{p}'(v) = \tilde{p}(v) = 1$. Damit $\bar{r}(v) = 1$ für einen Knoten v gilt, muss nach Definition der Normalisierung $r(v) = 1$ oder $r(v) = l(v) \in \mathbb{N} \setminus \{1\}$ gelten. Der zweite Fall kann aber in $G(w')$ nicht auftreten, da nach Definition 1 sonst r vor v mit p übereinstimmen müsste. Also gilt an solchen Knoten $\tilde{p}(v) = 1 = r(v)$. Ist $\bar{p}'$ nun aufgrund einer Unterscheidung nach einem Konkatenationsknoten kleiner als $\bar{r}$, so gilt dies auch für $\tilde{p}$ und r und es ist nichts mehr zu zeigen. Unterscheiden sie sich an einem π-Knoten v_0, so gilt dort $\tilde{p}(v_0) = 1 = \bar{p}'(v_0) <_{3+3^*} \bar{r}(v_0)$. Damit muss $r(v_0) \in -\mathbb{N}$ sein, da sonst nach Definition der Normalisierung $r(v_0) \in \mathbb{N} \setminus \{1\}$ gelten müsste, was aber nicht möglich ist, wie oben ausgeführt ist. Also ist $\tilde{p}(v_0) <_{\omega+\omega^*} r(v_0)$

und damit $\tilde{p} <_{\omega+\omega^*,w'} r$ und auch $\tilde{p} <_{\omega+\omega^*,w} r$.

b) Gilt $v_{\bar{p}} \neq v_{\bar{p}'}$, so handelt es sich dabei um das linke und das rechte Kind des selben Konkatenationsknotens. Da $u_p = v_{\bar{p}}$ gilt, bleibt für $\bar{r}$ bzw. r keine andere Möglichkeit, als dass $u_r = v_{\bar{p}'}$ gilt. Nun lässt sich für den Unterbaum von $G(w)$, dessen Wurzel u_r ist, die Argumentation von oben anwenden. Damit ist $\tilde{p} <_{\omega+\omega^*,w} r$.

□

Lemma 13: Sei $(l,p,r) \in \bar{P}_{\omega+\omega^*}(w)$ für einen π-Term w über dem Alphabet Σ und sei $a \in \Sigma$.
Es gilt:

$$\overline{{}^{[\![w]\!]_{\omega+\omega^*}} {X_a}^{\mathrm{c}}(l,p,r)} = {}^{[\![w]\!]_{3+3^*}} {X_a}^{\mathrm{c}}\overline{(l,p,r)}$$
$$\overline{{}^{[\![w]\!]_{\omega+\omega^*}} {Y_a}^{\mathrm{c}}(l,p,r)} = {}^{[\![w]\!]_{3+3^*}} {Y_a}^{\mathrm{c}}\overline{(l,p,r)}$$

Beweis. Nach den beiden vorangegangen Lemmata ist es unmöglich, dass eine der beiden Seiten der Gleichungen definiert, die andere aber undefiniert ist. Interessant ist hier nur der Fall, dass beide von ihnen definiert sind. Wieder wird hier aufgrund der Dualität nur der Fall für X_a betrachtet.

Um einen besseren Überblick behalten zu können sollen die folgenden Bezeichnungen gelten:

$$\overline{(l,p,r)} = (\bar{l},\bar{p},\bar{r}) \qquad {}^{[\![w]\!]_{3+3^*}} {X_a}^{\mathrm{c}}(\bar{l},\bar{p},\bar{r}) = (\bar{l}',\bar{p}',\bar{r}')$$
$$\overline{(\bar{l}',\bar{p}',\bar{r}')} = (\bar{\bar{l}},\bar{\bar{p}},\bar{\bar{r}}) \qquad {}^{[\![w]\!]_{\omega+\omega^*}} {X_a}^{\mathrm{c}}(l,p,r) = (l',p',r')$$
$$\overline{(l',p',r')} = (\tilde{l},\tilde{p},\tilde{r})$$

Man beachte, dass nach Definition der kondensierten Ranker $l' = p$, $r' = r$, $\bar{l}' = \bar{p}$ und $\bar{r}' = \bar{r}$ gilt. Es ist nun zu zeigen, dass $\bar{\bar{l}} = \tilde{l}$, $\bar{\bar{p}} = \tilde{p}$ und $\bar{\bar{r}} = \tilde{r}$ gilt. Dabei kann angenommen werden, dass $\bar{\bar{r}} \neq +\infty$ und $\tilde{r} \neq +\infty$ gilt. Ansonsten müsste $r = +\infty$ und $\bar{r} = +\infty$ gelten, woraus die entsprechende Aussage sofort folgen würde.

Um dies zu zeigen ist jedoch zunächst eine andere Aussage hilfreich. Sei u_0 der erste Knoten auf p, an dem sich p und p' unterscheiden; sei v_0 der entsprechende Knoten auf p'. Sei $\bar{u}_0$ der erste Knoten auf $\bar{p}$, an dem sich $\bar{p}$ und $\bar{p}'$ unterscheiden; sei $\bar{v}_0$ der entsprechende Knoten auf $\bar{p}'$. Nun gilt: p' und $\bar{p}'$ verlaufen über die selben Knoten und es ist $v_0 = \bar{v}_0$. Um dies einzusehen ist eine Fallunterscheidung notwendig. Liegt u_0 vor $\bar{u}_0$ auf p bzw. $\bar{p}$, so lässt sich ein Pfad q konstruieren. Dieser verläuft über die selben Knoten wie $\bar{p}'$ und hat vor $\bar{v}_0$ die selben Werte wie p. Ist $\bar{v}_0 = \bar{u}_0$ und damit ein π-Knoten, hat q dort den Wert $p(\bar{v}_0) + 1$. Dies ist möglich, da $p(\bar{v}_0) \neq -1$ sein muss. Andernfalls hätte dort auch $\bar{p}$ den Wert -1 und $\bar{p}'$ könnte keinen größeren Wert haben. An den anderen π-Knoten hat q den Wert 1. Damit ist $p <_{\omega+\omega^*,w} q <_{\omega+\omega^*} p'$ und q endet in einem mit a beschrifteten Knoten. Dies ist ein Widerspruch zu Minimalität von p'. Liegt $\bar{u}_0$ vor v_0, lässt sich ein ähnlicher Pfade konstruieren, der im Widerspruch zur Minimalität von $\bar{p}'$ steht. Dabei ist einzig zu beachten, dass für den Fall $v_0 = u_0$ nicht $\bar{p}(v_0) = -1$ gelten kann. Sonst wäre dort $p(v_0) = -1$ oder es würde $p(v_0) = r(v_0)$ gelten, beides ist nicht möglich, da sonst p' keinen größeren Wert haben könnte. Also muss $u_0 = \bar{u}_0$ gelten. Der einzige Fall, in dem trotzdem $v_0 \neq \bar{v}_0$ gelten kann, ist der folgende: $u_0 = \bar{u}_0$ ist ein π-Knoten und das linke Kind eines Konkatenationsknotens, dabei ist v_0 oder $\bar{v}_0$ das andere Kind und der verbleibende Knoten ist gleich zu u_0. Man verifiziere, dass sich in diesem Fall allerdings analog zu oben ein Pfad q im Widerspruch zur Minimalität von p' oder $\bar{p}'$ konstruieren lässt – die Konstruktion ist dabei sogar gleich. Es bleibt festzuhalten, dass also $v_0 = \bar{v}_0$ gelten muss. Da nach diesem Knoten beide Pfade, p' und $\bar{p}'$, an π-Knoten nur noch den Wert 1 annehmen können, würde aus einem unterschiedlichen weiteren Verlauf wieder ein Widerspruch zur Minimalität entstehen.

Damit verlaufen p' und $\bar{p}'$ über die selben Knoten, womit auch $\tilde{p}$ und $\bar{\bar{p}}$ über die selben Knoten verlaufen. Leicht überlegt man sich, dass $\tilde{l}$ und $\bar{\bar{l}}$ sowie $\tilde{r}$ und $\bar{\bar{r}}$ ebenfalls jeweils über die selben Knoten verlaufen. Es ist also nur noch zu zeigen, dass die Pfade in den Werten übereinstimmen. Sei also v ein π-Knoten aus dem gemeinsamen Anfang von p und p'. Wie bereits gezeigt ist v damit auch aus dem gemeinsamen Anfang von $\bar{p}$ und $\bar{p}'$. Die Gleichheit ergibt sich durch Rechnungen in den einzelnen Fällen, dabei gilt jedoch immer $p(v) = p'(v) = l'(v)$ und $\bar{p}(v) = \bar{p}'(v)$:

- $p(v) = 1$ bzw. $= -1$: Direkt aus der Definition ergibt sich $\tilde{p}(v) = \bar{p}(v) = \bar{\bar{p}}(v) = \tilde{l}(v) = \bar{\bar{l}}(v) = 1$ bzw. $= -1$.
- $p(v) \in \mathbb{N}, p(v) > 1$: Es ist $\tilde{p}(v) = p'(v) - l'(v) + 1 = p(v) - p(v) + 1 = 1$. Nach Definition ist außerdem $\tilde{l}(v) = \bar{\bar{l}}(v) = 1$. Schließlich ist $\bar{p}(v) = p(v) - l(v) + 1 \in \mathbb{N}$ und es muss unterschieden werden:
  - $\bar{p}(v) = 1$: $\bar{\bar{p}}(v) = 1$.
  - $\bar{p}(v) > 1$: $\bar{\bar{p}}(v) = \bar{p}'(v) - \bar{l}'(v) + 1 = \bar{p}(v) - \bar{p}(v) + 1 = 1$.
- $p(v) \in -\mathbb{N}, p(v) < -1$: Es ist $\tilde{p}(v) = p'(v) - r'(v) - 1 = p(v) - r(v) - 1 = \bar{p}(v)$. Unterscheide daher:
  - $p(v) - r(v) - 1 = -1$: $\bar{\bar{p}}(v) = -1$, $\tilde{l}(v) = l'(v) - r'(v) - 1 = p(v) - r(v) - 1 = -1$ und $\bar{\bar{l}}(v) = \bar{l}'(v) - \bar{r}'(v) - 1 = \bar{p}(v) - \bar{r}(v) - 1 = -1 - (-1) - 1 = -1$.
  - $p(v) - r(v) - 1 < -1$: $\bar{\bar{p}}(v) = \bar{p}'(v) - \bar{r}'(v) - 1 = p(v) - r(v) - 1 - \bar{r}(v) - 1 = -1$, da nach Definition $\bar{r}(v) = -1$, $\tilde{l}(v) = -1$ und $\bar{\bar{l}}(v) = -1$.

Damit stimmen $\tilde{p}$ und $\bar{\bar{p}}$, sowie $\tilde{l}$ und $\bar{\bar{l}}$ vor v_0 überein. Nach v_0 haben $\tilde{p}$ und $\bar{\bar{p}}$ ohnehin überall den Wert 1. Ist $v_0 \neq u_0$ ist damit für $\tilde{p}$ und $\bar{\bar{p}}$ nichts mehr zu zeigen. Ist $v_0 = u_0$ und damit ein π-Knoten, so gilt $p'(v_0) = p(v_0) + 1$ und $\bar{p}'(v_0) = \bar{p}(v_0) + 1$. Wieder ist eine Rechnung in den einzelnen Fällen nötig:

- $p(v_0) \in \mathbb{N}$: Dann ist $p'(v_0), \bar{p}'(v_0) > 1$ und $\tilde{p}(v_0) = p'(v_0) - l'(v_0) + 1 = p(v_0) + 1 - p(v_0) + 1 = 2$, sowie $\bar{\bar{p}}(v_0) = \bar{p}'(v_0) - \bar{l}'(v_0) + 1 = \bar{p}(v_0) + 1 - \bar{p}(v_0) + 1 = 2$. Ferner ist $\tilde{l}(v_0) = 1$ und $\bar{\bar{l}}(v_0) = 1$.
- $p(v_0) = -1$: Dieser Fall tritt nicht auf.
- $p(v_0) = -2$: Dann ist $p'(v_0) = -1$ und auch $\tilde{p}(v_0) = -1$. Man beachte, dass $r(v_0) > -2$ und damit $r(v_0) = -1$ gelten muss, also auch $\bar{r}(v_0) = -1$. Es ist $\bar{p}(v_0) = p(v_0) - r(v_0) - 1 = -2$ und somit $\bar{p}'(v_0) = -1$ und $\bar{\bar{p}}(v_0) = -1$. Schließlich ist $\tilde{l}(v_0) = l'(v_0) - r'(v_0) - 1 = p(v_0) - r(v_0) - 1 = -2$ und $\bar{\bar{l}}(v_0) = \bar{l}'(v_0) - \bar{r}'(v_0) - 1 = \bar{p}(v_0) - \bar{r}(v_0) - 1 = -2$.
- $p(v_0) \in -\mathbb{N}, p(v_0) < -2$: Es ist $p'(v_0) < -1$ und $r(v_0) = p(v_0) + 1$, da sonst $p'(v_0)$ größer als $r(v_0)$ wäre. Somit ist $\tilde{p}(v_0) = p'(v_0) - r'(v_0) - 1 = p(v_0) + 1 - r(v_0) - 1 = -1$, $\bar{p}(v_0) = p(v_0) - r(v_0) - 1 = -2$ und $\bar{p}'(v_0) = -1 = \bar{\bar{p}}(v_0)$. Schließlich gilt $\tilde{l}(v_0) = l'(v_0) - r'(v_0) - 1 = p(v_0) - r(v_0) - 1 = -2$ und $\bar{\bar{l}}(v_0) = \bar{l}'(v_0) - \bar{r}'(v_0) - 1 =$

$\bar{p}(v_0) - \bar{r}(v_0) - 1 = -2 + 1 - 1 = -2$.
Dies zeigt die Gleichheit von $\tilde{p}$ und $\bar{\bar{p}}$. Für die Gleichheit von $\tilde{l}$ und $\bar{\bar{l}}$ fehlt nur noch der Bereich nach u_0 und an u_0, falls $u_0 \neq v_0$ aber trotzdem ein π-Knoten ist. Sei v aus diesem Bereich auf $\tilde{l}$ bzw. $\bar{\bar{l}}$ bzw. p. Wieder sind einige Rechnungen notwendig.

- $p(v) = 1$ bzw. $= -1$: $\tilde{l}(v) = \bar{p}(v) = \bar{\bar{l}}(v) = 1$ bzw. $= -1$.
- $p(v) \in \mathbb{N}, p(v) > 1$: Es ist $\tilde{l}(v) = 1$ und $\bar{p}(v) \in \mathbb{N}$. Nach Definition ist $\bar{\bar{l}}(v) = 1$.
- $p(v) \in -\mathbb{N}, p(v) < -1$: Es ist $\tilde{l}(v) = l'(v) - r'(v) - 1 = p(v) - r(v) - 1 = \bar{p}(v)$. Unterscheide weiter:
  - $\bar{p}(v) = -1$: $\bar{\bar{l}}(v) = -1$.
  - $\bar{p}(v) < -1$: Nach Definition 1 bleibt als einziger Wert $\bar{p}(v) = -2$. Weil $r(v) \in -\mathbb{N}$ ist, ist $\bar{r}(v) = -1$. Also: $\bar{\bar{l}}(v) = \bar{l}'(v) - \bar{r}'(v) - 1 = \bar{p}(v) - \bar{r}(v) - 1 = -2$.

Zum Abschluss des Beweises bleibt nur noch zu zeigen, dass die Werte von $\tilde{r}$ und $\bar{\bar{r}}$ an den π-Knoten übereinstimmen. Sei dazu v ein π-Knoten aus r bzw. $\bar{r}$. Ist $r(v) = 1$, so ist $\bar{r}(v) = 1 = \bar{r}'(v) = \bar{\bar{r}}(v) = r'(v) = \tilde{r}(v)$. Ist $r(v) \in -\mathbb{N}$, so ist $\bar{r}(v) = -1 = \bar{r}'(v) = \bar{\bar{r}}(v) = \tilde{r}(v)$. Dies folgt jeweils direkt aus der Definition der Normalisierung. Ist $r(v) \in \mathbb{N}$ und $r(v) > 1$, so bedarf es einer letzten Fallunterscheidung. Sets gilt jedoch $\tilde{r}(v) = r'(v) - l'(v) + 1 = r(v) - p(v) + 1$ und $\bar{r}(v) = r(v) - l(v) + 1$. Außerdem ist zu beachten, dass l, p, und r vor v nach Definition 1 übereinstimmen müssen. Es ist also $l(v) \leq p(v) \leq r(v)$ und ohnehin $r(v) - l(v) \in \{0, 1\}$.

- $r(v) = l(v)$: Dann ist sogar $l(v) = p(v) = r(v)$, weil die Pfade vor v übereinstimmen müssen. Damit ist $\tilde{r}(v) = 1$, $\bar{r}(v) = 1$ und schließlich $\bar{\bar{r}}(v) = 1$.
- $r(v) = p(v) = l(v) + 1$: Dann ist $\tilde{r}(v) = 1$, $\bar{r}(v) = 2$ und $\bar{p}(v) = p(v) - l(v) + 1 = 2$. Also ist $\bar{\bar{r}}(v) = \bar{r}'(v) - \bar{l}'(v) + 1 = \bar{r}(v) - \bar{p}(v) = 1$.
- $r(v) = p(v) + 1 = l(v) + 1$: Dann ist $\tilde{r}(v) = 2$ und (in beiden möglichen Fällen) $\bar{p}(v) = 1$. Also ist $\bar{\bar{p}}(v) = \bar{r}(v) - \bar{p}(v) = 2$.

□

Nach diesen recht umfangreichen Vorarbeiten, ist es nun möglich das folgende Theorem zu beweisen.

Theorem 4: Sei w eine π-Term über dem Alphabet Σ und sei $r = Z_1 Z_2 \ldots Z_n$ ein fester Ranker mit $Z_i \in Z_\Sigma$ für $i = 1, 2, \ldots, n$. Es sei $(l_1, p_1, r_1) = \overline{{}^3 Z_1{}^{\mathrm{c}}(\pm\infty)}$ und $(l_{i+1}, p_{i+1}, r_{i+1}) = \overline{{}^3 Z_{i+1}{}^{\mathrm{c}}(l_i, p_i, r_i)}$ für $i = 1, 2 \ldots, n-1$, falls die rechten Seiten definiert sind. Ansonsten seien die Werte gleich $\perp$.
Dann gilt:

$$ {}^{\omega} r^{\mathrm{c}}(\pm\infty) \neq \perp \iff (l_n, p_n, r_n) \neq \perp $$

Beweis. Tatsächlich gilt sogar die folgende Aussage, die per Induktion gezeigt werden soll:

$$ \overline{{}^{\omega} r^{\mathrm{c}}(\pm\infty)} = (l_n, p_n, r_n) $$

Man beachte, dass dies das Gewünschte zeigt.
Zunächst ist $\overline{{}^3 Z_1{}^{\mathrm{c}}(\pm\infty)} = \overline{{}^{\omega} Z_1{}^{\mathrm{c}}(\pm\infty)}$. Damit ist auch $(l_1, p_1, r_1) = \overline{{}^3 Z_1{}^{\mathrm{c}}(\pm\infty)} = \overline{{}^{\omega} Z_1{}^{\mathrm{c}}(\pm\infty)}$, was den Induktionsanfang bildet.

Sei nun $1 \leq i < n$ und $\overline{{}^{\omega}Z_1 Z_2 \dots Z_i{}^{\mathrm{c}}(\pm\infty)} = (l_i, p_i, r_i)$. Es ist mit Lemma 13:

$$\begin{aligned}
\overline{{}^{\omega}Z_1 Z_2 \dots Z_i Z_{i+1}{}^{\mathrm{c}}(\pm\infty)} &= \overline{{}^{\omega}Z_{i+1}{}^{\mathrm{c}}({}^{\omega}Z_1 Z_2 \dots Z_i{}^{\mathrm{c}}(\pm\infty))} \\
&= \overline{{}^{3}Z_{i+1}{}^{\mathrm{c}}\left(\overline{{}^{\omega}Z_1 Z_2 \dots Z_i{}^{\mathrm{c}}(\pm\infty)}\right)} \\
&= \overline{{}^{3}Z_{i+1}{}^{\mathrm{c}}(l_i, p_i, r_i)} \\
&= (l_{i+1}, p_{i+1}, r_{i+1})
\end{aligned}$$

Besonderes Augenmerk sollt auch darauf gerichtet werden, dass dies auch gilt, wenn eine der beiden Seiten undefiniert bzw. $= \perp$ ist. □

Korollar: Sei $m \geq 1$.
Das Wortproblem für π-Terme von $\boldsymbol{R_m}$, das von $\boldsymbol{L_m}$ und das von $\boldsymbol{R_m} \vee \boldsymbol{L_m}$ ist entscheidbar. Außerdem ist das Wortproblem für π-Terme von **DA** entscheidbar.

Beweis. Nach Theorem 3 reicht es zum Lösen des Wortproblems für π-Terme der Varietäten aus die beiden Eingabeterme auf Unterscheidbarkeit durch gewisse kondensierte Ranker zu testen. Bezüglich der Definiertheit macht es nach Theorem 4 keinen Unterschied, ob X_a^{c} direkt angewendet wird oder ob erst normalisiert, dann X_a^{c} angewendet und schließlich erneut normalisiert wird. Selbiges gilt natürlich auch für Y_a^{c}. Nach der Normalisierung sind die auftretenden Pfade stets $(2+2^*)$-Pfade, wovon es allerdings in beiden Wörtern nur endlich viele gibt. Es lässt sich damit für jedes Wort ein (endlicher) Graph konstruieren, dessen Knoten die möglichen Werte nach iterierter Anwendung eines kondensierten Rankers der Länge 1 und anschließender Normalisierung darstellen. Die Kanten lassen sich mit dem jeweiligen Ranker beschriften. Anschließend muss nur noch überprüft werden, ob es eine Abfolge von Rankern aus der zur Varietät gehörenden Ranker-Menge gibt, so dass im einen Graphen $\perp$, im anderen Graphen jedoch ein definierter Wert erreicht wird. □

3 Parallelisierbarkeit

Die Wortprobleme für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie, sowie das von $\mathbf{DA}$ sind also entscheidbar. Doch wie effizient lassen sich diese Probleme lösen? Da sich weder Zeit- noch Platzkomplexität des am Ende des letzten Kapitels skizzierten Algorithmus einfach bestimmen lassen, widmet sich dieses Kapitel der Entwicklung eines Algorithmus für diese Probleme, der sich effizient parallel ausführen lässt.

Was dies genau bedeutet, klärt der erste Abschnitt dieses Kapitels. Gleichzeitig wird das grobe Vorgehen erläutert. Der nächste Abschnitt geht anschließend genauer auf den wichtigsten Punkt dieses Vorgehens ein und im letzten Abschnitt wird alles zu einem Ganzen zusammengesetzt.

3.1 Zielsetzung und Grundgerüst

3.1.1 Zielsetzung

Oft wird „Nick's Class" NC als die Klasse der parallel effizient entscheidbaren Probleme angesehen. Die Probleme in NC lassen sich durch einen Schaltkreis polynomieller Größe und mit polylogarithmischer Tiefe entscheiden.[†] Statt die Zugehörigkeit der Wortprobleme für π-Terme der hier besprochenen Varietäten direkt – beispielsweise durch Konstruktion der entsprechenden Schaltkreise – zu zeigen, soll hier ein anderer Weg eingeschlagen werden: Es sollen Algorithmen angegeben werden, die zur Zugehörigkeit der Probleme zu NL führen. Da NL Teilmenge von NC ist (vgl. z. B. [18]), zeigt dies das Gewünschte. Durch Determinierung der Algorithmen ergeben sich außerdem deterministische Polynomialzeit-Algorithmen zur Lösung der Probleme.

3.1.2 Grundgerüst

Der NL-Algorithmus zur Entscheidung des Wortproblems für π-Terme einer Varietät $\boldsymbol{V} \in \{\mathbf{DA}, \boldsymbol{R_m}, \boldsymbol{L_m}, \boldsymbol{R_m} \vee \boldsymbol{L_m} : m \in \mathbb{N}\}$ geht folgendermaßen vor: Es sei Σ das gemeinsame endliche Alphabet der beiden Eingabe-π-Terme u und v. Nach Theorem 3 gibt es eine Menge von kondensierten Rankern, sodass $u = v$ in $\boldsymbol{V}$ genau dann nicht gilt, wenn ein kondensierter Ranker aus der Menge auf $[\![u]\!]_{\omega+\omega^*}$ oder $[\![v]\!]_{\omega+\omega^*}$ definiert, auf dem jeweils anderen aber undefiniert ist. Um diese Unterscheidbarkeit zu prüfen wird zu jedem der beiden π-Terme eine Konfiguration[‡] gespeichert. Im Wesentlichen handelt es sich dabei um ein normalisiertes Tripel von $(3 + 3^*)$-Pfaden in $G(u)$ bzw. $G(v)$. Am Anfang ist dies für beide Terme $\pm\infty$. Nun wird schrittweise nichtdeterministisch ein

[†] Eine genaue Definition von NC findet sich z. B. in [18].

[‡] Diese Konfiguration ist nicht zu verwechseln mit der Konfiguration der Maschine selbst!

Element Z aus Z_Σ geraten, sodass die Abfolge der auf diesem Berechnungspfad geratenen Ranker ein Element aus der oben erwähnten Menge kondensierter Ranker ist. Nach dem Raten wird ${}^{[\![u]\!]_{3+3^*}}Z^{\mathrm{c}}$ auf das durch die Konfiguration in u codierte Element aus $\bar{P}_{\omega+\omega^*}(u)$ angewendet. Das Ergebnis wird dabei direkt normalisiert und codiert. Das Gleiche geschieht in jedem Schritt anschließend auch für v. Die Konfiguration codiert also die Werte der (l_i, p_i, r_i) aus Theorem 4. Lässt sich nun aus den beiden Konfigurationen schließen, dass die Anwendung des kondensierten Rankers bei u oder v definiert, beim jeweils anderen aber undefiniert war, so bedeutet dies, dass sich $[\![u]\!]_{\omega+\omega^*}$ und $[\![v]\!]_{\omega+\omega^*}$ durch einen kondensierten Ranker aus der zu $\boldsymbol{V}$ gehörenden Menge unterscheiden lassen. Der Ranker ergibt sich dabei aus dem Berechnungspfad der Maschine. In diesem Fall, akzeptiert die Maschine. Kann kein weiterer Ranker geraten werden, bricht die Maschine ohne Akzeptanz ab.

Die Maschine akzeptiert also genau dann, wenn $u = v$ in $\boldsymbol{V}$ *nicht* gilt. Ziel ist es nun zu zeigen, dass die Maschine nur logarithmisch viel Platz in der Eingabelänge benötigt. Gelingt dies, so lässt sich nach Abschluss nichtdeterministischer Platzklassen unter Komplement in NL entscheiden, ob $u = v$ in $\boldsymbol{V}$ gilt. Zum Raten von Z wird ein einzelnes Bit benötigt; betrachtet man Σ als Eingabe muss auch ein einzelner Buchstabe gespeichert werden, dies ist aber in logarithmischem Platz in der Größe von Σ möglich. Für $\boldsymbol{V} \in \{\boldsymbol{R_m}, \boldsymbol{L_m}, \boldsymbol{R_m} \vee \boldsymbol{L_m} : m \in \mathbb{N}\}$ muss zudem die Anzahl der noch möglichen Wechsel zwischen X_Σ und Y_Σ gespeichert werden, dabei handelt es sich um eine Zahl zwischen 0 und m. Sofern m nicht als Eingabe betrachtet wird, ist dafür nur konstant viel Platz notwendig. Schließlich muss noch eine Möglichkeit gefunden werden die Konfigurationen für u und v zu speichern. Dabei ist darauf zu achten, dass dies in logarithmischen Platz erfolgt und dass sich die Nachfolgekonfiguration – also die Anwendung eines kondensierten Rankers mit anschließender Normalisierung – ausrechnen lässt ohne den logarithmisch beschränkten Platz zu verlassen. Tatsächlich bedarf dies jedoch einer genaueren Betrachtung, die im nächsten Abschnitt erfolgt.

3.2 Effiziente Speicherung

3.2.1 Grundlegende Ideen

Die Speicherung der Konfigurationen für die beiden Eingabe-π-Terme kann unabhängig von der jeweils anderen erfolgen ohne die Platzbedingung zu verletzten. Daher soll für diesen Abschnitt ein π-Term w über dem Alphabet Σ fixiert werden. Die Ergebnisse zur Speicherung der Konfiguration von w lassen sich dann später jeweils auf beide Terme einzeln anwenden.

Bevor weitere Aussagen zur Speicherung der Konfiguration von w gemacht werden können, muss zunächst geklärt werden, welche Information in ihr codiert werden muss: Anfangs muss der Wert $\pm\infty$ gespeichert werden. Nach Anwendung eines Rankers und Normalisierung ergibt sich dann ein normalisierter Wert aus $\bar{P}_{\omega+\omega^*}(w)$ oder direkt $\perp$. Da weitere Möglichkeiten nicht auftreten, kann die Information, in welchem der drei Fälle sich die Konfiguration befindet, in konstantem Platz gespeichert werden. Von weiterem Interesse ist dann nur noch die Speicherung eines normalisierten Elements aus $\bar{P}_{\omega+\omega^*}(w)$.

Hierbei handelt es sich um ein Tripel (l,p,r), wobei $l = -\infty$ und $r = +\infty$ sein kann, es sich ansonsten aber um $(3+3^*)$-Pfade in $G(w)$ handelt. Problematisch ist dabei nur die Speicherung der Werte an den π-Knoten im Syntax-Baum, da sich der Verlauf eines Pfades durch ein Blatt speichern lässt. Bei den Blättern handelt es sich um Positionen im π-Term w, die nur logarithmisch viel Platz in der Länge von w benötigen. Zunächst wäre es nun denkbar, beispielsweise mithilfe eines zusätzlichen Bandes, unter jedem $^\pi$ den Wert von l, p und r zu speichern. Obwohl die Werte nur aus $\{1,2,-2,-1\}$ stammen können, würde dies jedoch bereits linear viel Speicher in der Länge von w benötigen, ist also für die Zwecke hier ungeeignet.

Eine erste Idee zur Reduktion dieses Platzbedarfs liefert die folgende Beobachtung: Verlaufen l, p und r am Anfang gemeinsam, so kann die Anwendung eines Rankers den Wert von p an den π-Knoten auf diesem gemeinsamen Anfang nicht mehr verändern. Ansonsten wäre die Zulässigkeit von (l,p,r) verletzt! Statt also die Werte an diesen π-Knoten zu speichern, ist es ausreichend eine einzige Position in w zu speichern. Die mit $^\pi$ beschrifteten Positionen rechts dieser Position können dann nicht mehr erreicht werden, weil sich r links von ihnen befindet oder die Werte aller drei Pfade dort übereinstimmen. Dies führt nun zwar zu einer Verringerung des Platzbedarfs, allerdings ist dieser im schlechtesten Fall immer noch linear in der Länge von w; beispielsweise wenn kein gemeinsamer Anfang existiert. Um auch eine asymptotische Verbesserung zu erzielen ist also mehr notwendig.

3.2.2 Ein letztes Lemma

Um die Menge an Information, die gespeichert werden muss, zu reduzieren ist es hilfreich mehr Erkenntnisse über das Tripel (l,p,r) zu gewinnen. Dazu dient dieser Unterabschnitt, der zunächst mit einer Definition beginnt:

Definition: Sei $(l,p,r) \in \bar{P}_{\omega+\omega^*}$ und $q \in \{l,p,r\}$. Dann bezeichne $\delta_q(l,p,r)$ den ersten Knoten auf q, an dem sich q von einem der beiden anderen Pfade unterscheidet.
Ferner heißt q in *X-Form*, falls für alle π-Knoten v auf q gilt: Liegt v nach $\delta_q(l,p,r)$ und gilt $q(v) \in \mathbb{N}$, so gilt $q(u) \in \mathbb{N}$ auch für alle π-Knoten u, die nach v auf q liegen. Schließlich heißt q in *Y-Form*, falls für alle π-Knoten v auf q gilt: Liegt v nach $\delta_q(l,p,r)$ und gilt $q(v) \in -\mathbb{N}$, so gilt $q(u) \in -\mathbb{N}$ auch für alle π-Knoten u, die nach v auf q liegen.

Anschaulich ist ein Pfad also in X-Form, wenn nach dem ersten Knoten hinter dem gemeinsamen Anfang der drei Pfade nach dem ersten Auftreten eines Wertes aus $\mathbb{N}$ alle weiteren Werte ebenfalls aus $\mathbb{N}$ sind. Die Y-Form ist dazu symmetrisch: Nach dem ersten Wert aus $-\mathbb{N}$ sind auch alle weiteren Werte aus $-\mathbb{N}$. Man beachte, dass nach Lemma 12 X- und Y-Form durch eine mögliche Normalisierung erhalten bleiben!

Lemma 14: Sei w ein beliebiger π-Term über Σ und $r = Z_1 Z_2 \ldots Z_n$ ein Ranker. Definiere $(l_1,p_1,r_1) := \overline{{}^{[\![w]\!]_{3+3^*}} Z_1{}^{\mathrm{c}}(\pm\infty)}$ und $(l_{i+1},p_{i+1},r_{i+1}) := \overline{{}^{[\![w]\!]_{3+3^*}} Z_{i+1}{}^{\mathrm{c}}(l_i,p_i,r_i)}$ für $i = 1,2,\ldots,n-1$. Für alle $i \in \{1,2,\ldots,n\}$ mit $(l_i,p_i,r_i) \neq \perp$ gilt:

- $l_i = -\infty$ oder l_i ist in X- oder in Y-Form,
- $r_i = +\infty$ oder r_i ist in X- oder in Y-Form und
- p_i ist in X-Form und alle π-Knoten auf p_i nach $\delta_{p_i}(l_i)$ haben Werte aus $\mathbb{N}$ oder p_i ist in Y-Form und alle π-Knoten auf p_i nach $\delta_{p_i}(r_i)$ haben Werte aus $-\mathbb{N}$.

Beweis. Ohne Einschränkung sei $(l_i, p_i, r_i) \neq \bot$ für alle $i = 1, 2, \ldots, n$. Für (l_1, p_1, r_1) gilt die Aussage direkt, da $l_1 = -\infty$, $r_1 = +\infty$ und p_1 an den π-Knoten überall den Wert 1 oder überall den Wert -1 hat.

Sei die Aussage nun für (l_i, p_i, r_i) gezeigt. Definiere die Bezeichnungen:

$$(l, p, r) = (l_i, p_i, r_i) \qquad (l', p', r') = {}^{[\![w]\!]_{3+3^*}} {Z_{i+1}}^{\mathrm{c}}(l, p, r)$$

Außerdem sei $v_p = \delta_p(p')$ und $v_{p'} = \delta_{p'}(p)$. Der Fall $Z_{i+1} \in Y_\Sigma$ verhält sich wieder dual zum hier besprochenen Fall $Z_{i+1} \in X_\Sigma$, in dem $r' = r$ gilt und damit für r' nichts mehr zu zeigen ist. Zunächst sei dabei p in X-Form und es gelte $p(v) \in \mathbb{N}$ für alle π-Knoten v auf p nach $\delta_p(l)$. Für $v_p \neq v_{p'}$ hat p' an π-Knoten ab $v_{p'}$ den Wert 1. Damit ist p' in X-Form und auch die Zusatzbedingung ist erfüllt. Zudem ist $l' = p$ nach Voraussetzung in X-Form. Ist $v_p = v_{p'}$, so handelt es sich um einen π-Knoten und es gilt $p'(v_p) = p(v_p) + 1$. Nach v_p hat p' an allen π-Knoten den Wert 1. Es verbleibt damit nur noch zu zeigen, dass p' in X-Form ist. Für $p(v_p) \in \mathbb{N}$ und damit auch $p'(v_p) \in \mathbb{N}$ ist demnach nichts mehr zu zeigen. Sei also $p(v_p) \in -\mathbb{N}$. Da $p'(v_p) = p(v_p) + 1$ gilt, muss $p(v_p) < -1$ sein. Nach Definition 1 ist dann $v_p = \delta_p(r)$. Vor v_p stimmen p, p' und r also überein, womit $\delta_{p'}(l', p', r') = v_p$ ist. Da p' an π-Knoten nach v_p überall den Wert 1 hat, ist p' somit in X-Form.

Nun sei p in Y-Form und es gelte $p(v) \in -\mathbb{N}$ für alle π-Knoten v auf p nach $\delta_p(r) =: v_r$. Läge v_p echt vor v_r, so wäre $p' >_{\omega+\omega^*,w} r = r'$. Da dies nicht möglich ist, kann v_p nur nach v_r auf p liegen oder es gilt $v_p = v_r$. Liegt v_p echt vor v_r, so kann der Fall $v_p = v_{p'}$ nicht mehr auftreten: Es müsste dann $p(v_p) \in -\mathbb{N}$ sein und außerdem wegen $p'(v_p) = p(v_p) + 1$ auch $p(v_p) < -1$ gelten. Nach Definition 1 wäre dann aber $v_p = v_r$. Also hat p' zunächst die selben Werte wie p, aber ab $v_{p'}$ tritt an π-Knoten nur noch der Wert 1 auf. Also ist p' in X-Form und auch die Zusatzbedingung ist erfüllt. Gilt $v_p = v_r$, so ist $\delta_{p'}(l', p', r') = v_{p'}$. Da nach $v_{p'}$ nur noch 1 als Wert an π-Knoten auf p' auftritt, ist p' in X-Form und auch die Zusatzbedingung ist erfüllt. Schließlich ist $l' = p$ in Y-Form.

Die gewünschten Eigenschaften gelten also für l', p' und r'. Da sie durch die Normalisierung erhalten bleiben, gelten sie damit auch für l_{i+1}, p_{i+1} und r_{i+1}. □

Was bedeutet dies nun für die Speicherung von (l_i, p_i, r_i)? Wie bereits gesehen, müssen die Werte eines Pfades $q \in \{l_i, p_i, r_i\}$ an den π-Knoten vor $\delta_q(l_i, p_i, r_i)$ nicht gespeichert werden. Nach $\delta_q(l_i, p_i, r_i)$ kann jedoch nur ein einziger Wechsel von Werten aus $\mathbb{N}$ zu Werten aus $-\mathbb{N}$ oder umgekehrt erfolgen. Kombiniert man dies mit den Forderungen aus Definition 1 und der Definition der Normalisierung, so ergibt sich Folgendes: l_i kann an π-Knoten nur Werte aus $\{1, -1, -2\}$ annehmen und r_i nur welche aus $\{1, 2, -1\}$. Zudem Kann der Wert -2 für l_i nur bei $\delta_{l_i}(l_i, p_i, r_i)$ auftreten und der Wert 2 für r_i nur bei $\delta_{r_i}(l_i, p_i, r_i)$. Ebenso ist der Wert 2 auf p_i dem Knoten $\delta_{p_i}(l_i)$ und der Wert -2 dem Knoten $\delta_{p_i}(r_i)$ vorbehalten. Dies bedeutet jedoch beispielsweise nicht, dass $p_i(\delta_{p_i}(l_i)) \neq -2$ sein muss! Dies liegt daran, dass $\delta_{p_i}(l_i) = \delta_{p_i}(r_i)$ sein kann. Die Werte an den anderen Knoten sind 1 oder -1. Somit reicht es für diese Knoten aus die Form des Pfades zu speichern. Für den Rest muss nur eine konstante Anzahl an Knoten mit den zugehörigen Werten gespeichert werden.

3.2.3 Die Konfiguration im Detail

Der letzte Unterabschnitt hat gezeigt, dass sich ein normalisiertes Element (l, p, r) aus $\bar{P}_{\omega+\omega^*}(w)$ in logarithmischem Platz speichern lässt. Dieser Unterabschnitt führt dies genauer aus und geht auch darauf ein, wie sich aus den gespeicherten Werten die Nachfolgekonfiguration errechnen lässt. Die folgende Tabelle fasst die Variablen zusammen, die dazu verwendet werden (l, p, r) zu speichern. Man beachte, dass diese tatsächlich im verfügbaren Platz gespeichert werden können.

Name	**Verwendung**	**Werte**	**Größe**
`start`	Ende des gemeinsamen Anfangs von l, p und r; für $^\pi$-Positionen links dieses Werts können die Werte aller drei Pfade berechnet werden; codiert im Wesentlichen $\delta_l(l,p,r)$, $\delta_p(l,p,r)$ und $\delta_r(l,p,r)$	Positionen in w und $\perp$	$\mathcal{O}(\log\|w\|)$
`l_form`	Form von l nach $\delta_l(l,p,r)$	$X, Y, -\infty$	$\mathcal{O}(1)$
`p_form`	Form von p nach $\delta_p(l,p,r)$	X, Y	$\mathcal{O}(1)$
`r_form`	Form von r nach $\delta_r(l,p,r)$	$X, Y, +\infty$	$\mathcal{O}(1)$
`l_leaf`	Ende von l	Positionen in w und $\perp$	$\mathcal{O}(\log\|w\|)$
`p_leaf`	Ende von p	Positionen in w	$\mathcal{O}(\log\|w\|)$
`r_leaf`	Ende von r	Positionen in w und $\perp$	$\mathcal{O}(\log\|w\|)$
`delta_l`	Position von $\delta_l(l,p,r)$, falls dies ein π-Knoten ist	$^\pi$-Position in w oder $\perp$	$\mathcal{O}(\log\|w\|)$
`delta_l_value`	Wert von l an $\delta_l(l,p,r)$	$1, -1, -2$	$\mathcal{O}(1)$
`delta_pl`	Position von $\delta_p(l)$, falls dies ein π-Knoten ist	$^\pi$-Position in w oder $\perp$	$\mathcal{O}(\log\|w\|)$
`delta_pl_value`	Wert von p an $\delta_p(l)$	$1, 2, -1, -2$	$\mathcal{O}(1)$
`delta_pr`	Position von $\delta_p(r)$, falls dies ein π-Knoten ist	$^\pi$-Position in w oder $\perp$	$\mathcal{O}(\log\|w\|)$
`delta_pr_value`	Wert von p an $\delta_p(r)$	$1, 2, -1, -2$	$\mathcal{O}(1)$
`delta_r`	Position von $\delta_r(l,p,r)$, falls dies ein π-Knoten ist	$^\pi$-Position in w oder $\perp$	$\mathcal{O}(\log\|w\|)$
`delta_r_value`	Wert von r an $\delta_r(l,p,r)$	$1, 2, -1$	$\mathcal{O}(1)$
`l_switch`	Position auf l, an der ein Wechseln von Werten aus $\mathbb{N}$ zu Werten aus $-\mathbb{N}$ oder umgekehrt stattfindet	$^\pi$-Position in w oder $\perp$	$\mathcal{O}(\log\|w\|)$

Name	**Verwendung**	**Werte**	**Größe**
p_switch	Position auf p, an der ein Wechseln von Werten aus $\mathbb{N}$ zu Werten aus $-\mathbb{N}$ oder umgekehrt stattfindet	${}^{\pi}$-Position in w oder $\perp$	$\mathcal{O}(\log \lvert w \rvert)$
r_switch	Position auf r, an der ein Wechseln von Werten aus $\mathbb{N}$ zu Werten aus $-\mathbb{N}$ oder umgekehrt stattfindet	${}^{\pi}$-Position in w oder $\perp$	$\mathcal{O}(\log \lvert w \rvert)$

Als Nächstes ist zu klären, ob aus diesen Werten die Werte der drei Pfade an den π-Knoten rekonstruiert werden können. Zunächst ist dabei festzuhalten, dass in logarithmischem Platz entschieden werden kann, ob ein Pfad, der nur über eine Position in w gegeben ist, über einen bestimmten π-Knoten verläuft. Es ist hierbei nämlich nur notwendig zu überprüfen, ob sich die Blattposition in w zwischen den zum π-Knoten gehörenden Klammern befindet. Dies ist möglich, indem die Zahl der öffnenden und schließenden Klammer gezählt wird. Es muss also nur eine durch $|w|$ beschränkte Zahl gespeichert werden, was durch Binärkodierung in logarithmischem Platz möglich ist. Um nun beispielsweise den Wert von p an einem π-Knoten, der durch die zugehörige ${}^{\pi}$-Position in w gegeben ist, zu berechnen muss zunächst überprüft werden, ob p über den Knoten verläuft. Ist dies nicht der Fall, kann direkt $\perp$ zurück gegeben werden. Ist dies der Fall, aber der Knoten liegt auf dem gemeinsamen Anfang von (l, p, r), also nicht links von `start`, so ist eine Bestimmung des Wertes nicht möglich. Wie oben aufgeführt ist dies allerdings auch nicht notwendig, es kann also sichergestellt werden, dass die Routine zur Bestimmung des Wertes von p nie für einen solchen Knoten aufgerufen wird. Für Knoten zu ${}^{\pi}$-Positionen links von `start` wird zunächst verglichen, ob es sich um den in `delta_pl` oder `delta_pr` gespeicherten Knoten handelt. Ist dies der Fall, kann der gespeicherte Wert gelesen werden. Ist dies nicht der Fall, muss unterschieden werden: Hat `p_form` den Wert „X“ und `p_switch` liegt nach dem fraglichen Knoten auf p oder ist undefiniert, so ist nach Lemma 14 der gesuchte Wert -1. Nach oder an `p_switch` ist der Wert 1 und für `p_form` $= Y$ sind die Werte vertauscht. Die Algorithmen zur Bestimmung der Werte von l und r verlaufen analog. Hierbei sind nur die speziellen Werte $-\infty$ und $+\infty$ zu beachten, es kann allerdings wieder an anderer Stelle verhindert werden, dass die Routinen für diese Spezialfälle aufgerufen werden.

Um den Gesamtalgorithmus zu vervollständigen bleibt nur noch zu erörtern, dass sich die Nachfolgekonfiguration in der oben angegebenen Form in logarithmischem Platz ausrechnen lässt. Die Nachfolgekonfiguration muss dabei ein normalisiertes Element (l', p', r') aus $\bar{P}_{\omega+\omega^*}(w)$ codieren, dass durch Anwendung eines Rankers X_a oder Y_a und anschließender Normalisierung aus der aktuellen Konfiguration hervorgeht. Für den Spezialfall, dass die aktuelle Konfiguration den Wert $\pm\infty$ codiert, ist dies recht einfach, schließlich muss nur das erste bzw. letzte a in w gesucht werden. Gibt es kein a in w, kann $\perp$ zurückgegeben werden. Ansonsten kann $l' = -\infty$ und $r' = +\infty$ direkt in der für die Form der beiden Pfade vorgesehenen Variablen gespeichert werden. Da p' entweder an allen π-Knoten den Wert 1 oder an allen den Wert -1 hat, kann auch dies einfach in `p_form` gespeichert werden. Codiert die aktuelle Konfiguration hingegen das normalisierte Element (l, p, r) aus $\bar{P}_{\omega+\omega^*}(w)$ bedeutet dies größeren Aufwand. Da das Vorgehen für Y_a ähnlich zu dem für X_a ist, wird hier nur letzteres genauer besprochen. Zunächst werden die folgenden Hilfsvariablen benötigt:

Name	**Verwendung**	**Werte**	**Größe**
`cur_leaf`	Blatt, an dem der aktuell betrachtete Pfad endet	Positionen in w	$\mathcal{O}(\log \lvert w \rvert)$
`branching`	$^\pi$-Position, an der sich der aktuelle Pfad zum ersten Mal von p unterscheidet	$^\pi$-Position in w oder $\perp$	$\mathcal{O}(\log \lvert w \rvert)$

Es ist anzumerken, dass diese Variablen alle wieder in logarithmischem Platz gespeichert werden können.

Der Algorithmus soll nun zunächst durch Pseudocode skizziert werden:

```
cur_leaf ← p_leaf
branching ← ⊥
while true do
    cur_leaf ← cur_leaf + 1
    if cur_leaf > |w| or cur_leaf ≥ start or pathNotSmallerThanR then
        return ⊥
    end if
    if w(cur_leaf) = ")" and w(cur_leaf + 1) = "π" then
        if branching = cur_leaf then              ▷ Klammer zum zweiten Mal erreicht?
            branching ← ⊥
        else if cur_leaf ≥ start then             ▷ π-Position im gemeinsamen Anfang?
            return ⊥
        else if branching = ⊥ and isOnP(cur_leaf) and p(cur_leaf) ≠ −1 then
                                                   ▷ Klammer von vorne testen?
            branching ← cur_leaf
            cur_leaf ← matchingLeft(cur_leaf)
        end if
    else if w(cur_leaf) = a then
        break
    end if
end while
Berechne und speichere neue Konfiguration         ▷ Hier ist stets w(cur_leaf) = a
```

Verbal beschrieben ist das Vorgehen das folgende: Beginne beim Ende von p und gehe schrittweise in w jeweils einen Buchstaben nach rechts. Diese Position wird in `cur_leaf` gespeichert und definiert den Verlauf des aktuellen Pfades. Wird ein a gefunden, so ist dies das nächste und damit das Ende von p'. Wird die Buchstaben-Kombination $)^\pi$ gefunden, so muss – falls p an dieser Stelle nicht den Wert -1 hat – an der zugehörigen linken Klammer weiter gesucht werden. Hier kommt die Variable `branching` ins Spiel: Sie speichert die Position dieser Abzweigung. Dies geschieht zum einen um eine Endlosschleife zu verhindern und zum anderen um die Werte des aktuellen Pfades an einem π-Knoten v, über den der aktuelle Pfad verläuft, zu bestimmen. Verläuft p nicht über v ist dies stets der Wert 1. Ist v der Knoten zu der in `branching` gespeicherten $^\pi$-Position, so ist der Wert $p(v)+1$. Liegt v auf p oberhalb vom zu `branching` gehörenden π-Knoten oder ist `branching` $= \perp$, ist der Wert gleich $p(v)$; unterhalb ist es wiederum 1. Aus den beiden Hilfsvariablen lässt sich also ohne Verlassen des logarithmischen Platzes der Wert des aktuellen Pfades an allen Knoten zu den $^\pi$-Positionen links von `start` bestimmten. Dies ist insbesondere nötig um die Routine PATHNOTSMALLERTHANR zu implementieren. Diese soll sicherstellen, dass der aktuelle Pfad immer kleiner als r ist. Dazu muss von rechts nach links beginnend bei `start` für alle $^\pi$-Positionen überprüft werden, dass, sofern r und der aktuelle Pfade beide über den zugehörigen π-Knoten verlaufen, die Werte des aktuellen Pfades nicht größer als die von r sind. Sind sie stets gleich, so muss zudem überprüft werden, ob `cur_leaf` links von `r_leaf` liegt. All dies ist jedoch in logarithmischem Platz möglich.

Einen letzten Punkt beschreibt der Pseudocode nur unzureichend, nämlich wie die Nachfolgekonfiguration tatsächlich ausgerechnet wird. Hierzu kann zunächst der neue Wert für `start` ausgerechnet werden: Ähnlich wie in PATHNOTSMALLERTHANR wird von rechts nach links an allen $^\pi$-Positionen überprüft, ob es einen Unterschied in den Werten von l', p' und r' gibt. Anschließend kann `start` auf die $^\pi$-Positionen zum letzten übereinstimmenden π-Knoten gesetzt werden. Ähnlich können auch die Werte für `delta_l`, `delta_pl`, `delta_pr` und `delta_r`, sowie für die zugehörigen `delta_*_value`-Variablen ermittelt werden. Wichtig ist hierbei einzig, dass der normalisierte Wert gespeichert wird, dieser lässt sich allerdings direkt ausrechnen. Die Form der Pfade und die Position eines eventuellen Wechsels der Werte zwischen $\mathbb{N}$ und $-\mathbb{N}$ kann ebenfalls durch einfaches Ablaufen der Pfade bestimmt werden, womit alle Variablen der Konfiguration abgehandelt sind.

Zum Ende des Unterabschnitts zur Berechnung der Nachfolgekonfigurationen sei noch Folgendes explizit hervorgehoben: Die Berechnung der Nachfolgekonfiguration benötigt den Nichtdeterminismus nicht; sie erfolgt also deterministisch in logarithmischem Platz!

3.3 Zusammensetzen der Bausteine

Der deterministische Algorithmus zur Berechnung der Nachfolgekonfigurationen lässt sich nun in das beschriebene nichtdeterministische Grundgerüst einsetzen und liefert so einen NL-Algorithmus zur Entscheidung, ob $u = v$ in $\boldsymbol{V}$ mit $\boldsymbol{V} \in \{\mathbf{DA}, \boldsymbol{R_m}, \boldsymbol{L_m}, \boldsymbol{R_m} \vee \boldsymbol{L_m} : m \in \mathbb{N}\}$ gilt. Da es sich hierbei um das zentrale Ergebnis dieser Arbeit handelt, soll es abschließend durch ein Theorem gewürdigt werden:

Theorem: Für alle $m \in \mathbb{N}$ lässt sich das Wortproblem für π-Terme von $\boldsymbol{R_m}$, von $\boldsymbol{L_m}$ und von $\boldsymbol{R_m} \vee \boldsymbol{L_m}$ durch einen nichtdeterministischen Algorithmus entscheiden, dessen Platzbedarf in

$$\mathcal{O}\left(\log|u| + \log|v| + \log|\Sigma| + \log m\right)$$

liegt, wobei u und v die beiden Eingabe-π-Terme über dem gemeinsamen Alphabet Σ sind.
Das Wortproblem für π-Terme von **DA** lässt sich ebenso durch einen nichtdeterministischen Algorithmus mit Platzbedarf in

$$\mathcal{O}\left(\log |u| + \log |v| + \log |\Sigma|\right)$$

entscheiden. Auch hier bezeichnen u und v die beiden Eingabe-π-Terme.

4 Ausblick und Zusammenfassung

4.1 Ausblick

Bisher wurden nur die Wortprobleme für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie untersucht. Dies war zwar ausreichend um auch eine Aussage über das Wortproblem für π-Terme von **DA** zu machen, es drängt sich aber die Frage nach der Entscheidbarkeit und ggf. der Komplexität der Wortprobleme für π-Terme der $\cap$-Ebenen auf. Das Hauptproblem hierbei ist, dass sich diese nicht durch kondensierte Ranker beschreiben lassen, sondern eine andere Form von Rankern benötigen[†]. Daher lassen sich die Ergebnisse nicht direkt übertragen und die Idee der Normalisierung müsste entsprechend adaptiert werden.

Eine andere Frage ist die, ob sich die besprochenen Wortprobleme für π-Terme sogar in einer noch kleineren Komplexitätsklasse befinden oder ob sie auf der anderen Seite möglicherweise NL-vollständig sind. In diesem Zusammenhang ist auch die folgende Überlegung interessant: Es ist möglich die Nachfolgekonfiguration deterministisch in logarithmischem Platz auszurechnen. Möglicherweise ist es auf ähnliche Weise auch möglich den Graphen, wie er im skizzierten Algorithmus am Ende von Kapitel 2 angedeutet ist, deterministisch in logarithmischem Platz auszurechnen. Dies käme einer Reduktion der Probleme auf das Problem der Äquivalenz endlicher deterministischer Automaten sehr nahe, was weitere Einblicke in die Vollständigkeit des Problems für bestimmte Komplexitätsklassen erlauben würde. Hier würde sich auch die Frage anschließen, welche Komplexität die Konstruktion des Graphen genau besitzt.

Außerdem wäre es auch wünschenswert die vorgestellten Beweise zu vereinfachen. Hier scheint Potential zu bestehen, da sich insbesondere bei den vor allem auf Rechnungen basierenden Beweisen viele argumentative Überlappungen feststellen lassen.

Eine weitere Verbesserungsmöglichkeit findet sich beim Algorithmus aus Kapitel 3. Wäre es hier möglich die auftretenden Konstanten zu verkleinern, so würde dies direkt zu einem effizienteren deterministischen Polynomialzeitalgorithmus führen. Möglicherweise wäre es so sogar möglich einen effizienteren Algorithmus als den in [10] beschriebenen zu finden und das dortige Ergebnis weitestgehend zu subsumieren. Betrachtet man die Tabellen über die zur Speicherung einer Konfiguration nötigen Variablen, wird schnell deutlich, dass hier einige Optimierungen möglich sind.

[†]Diese Form der Ranker heißt in [7] *Weis-Immerman-Ranker*

4.2 Zusammenfassung

Nach der Einführung wichtiger Grundbegriffe im ersten Kapitel ist es gelungen das Konzept der Interpretation von $^\pi$-Potenzen in π-Termen durch lineare Ordnungstypen mit dem Konzept der Ranker und kondensierten Ranker zu vereinbaren. Dies führte darauf, dass sich die Gültigkeit einer Gleichung in den Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie auf die Unterscheidbarkeit der involvierten π-Terme durch gewisse kondensierte Ranker zurückführen lässt. Im Wesentlichen bedeutet dieses Ergebnis die Übertragung des Zusammenhangs zwischen Rankern und den Varietäten der Trotter-Weil-Hierarchie auf π-Terme und damit ins Unendliche. Daraus ergab sich direkt eine entsprechende Aussage für die Varietät $\mathbf{DA}$.

Der nächste Schritt auf dem Weg zur Entscheidbarkeit des Wortproblems für π-Terme der besprochenen Varietäten war die Definition einer Normalisierung eines Tripels auf $(\omega + \omega^*)$-Pfaden, die nach Ergebnissen aus dem ersten Kapitel den Positionen in dem verallgemeinerten Wort entsprechen, das durch Einsetzen des linearen Ordnungstyps $\omega + \omega^*$ für die $^\pi$-Potenzen eines π-Terms entsteht. Dies erlaubte die Beschränkung auf eine begrenzte Menge möglicher Werte und damit den Schritt zurück ins Endliche, was letztlich zum Beweis der Entscheidbarkeit und der Skizzierung eines entsprechenden Algorithmus führte.

Anschließend wurden logarithmisch im Platz beschränkte nichtdeterministische Algorithmen zur Entscheidung der Wortprobleme für π-Terme der Ecken und $\vee$-Ebenen der Trotter-Weil-Hierarchie sowie von $\mathbf{DA}$ präsentiert. Entscheidend dabei war die effiziente Speicherung eines normalisierten Tripels von $(\omega + \omega^*)$-Pfaden, die gleichzeitig das Ausrechnen einer Nachfolgekonfiguration in logarithmischem Platz erlaubt. Dies zeigte die Zugehörigkeit der Probleme zu NL und verbesserte damit hinsichtlich der Komplexitätsklasse ein aus [10] bereits bekanntes Ergebnis über die Entscheidbarkeit des Wortproblems für π-Terme von $\mathbf{DA}$ durch einen deterministischen Polynomialzeitalgorithmus.

Literatur

[1] Jorge Almeida und Marc Zeitoun. „An automata-theoretic approach to the word problem for ω-terms over R“. In: *Theoretical Computer Science* 370.1 (2007), S. 131–169.

[2] Jorge Almeida und Marc Zeitoun. „The equational theory of ω-terms for finite R-trivial semigroups“. In: *Proceedings of the Workshop Semigroups and Languages.* World Scientific, 2004.

[3] Volker Diekert, Paul Gastin und Manfred Kufleitner. „A survey on small fragments of first-order logic over finite words“. In: *International Journal of Foundations of Computer Science* 19.03 (2008), S. 513–548.

[4] John M. Howie. *Fundamentals of Semigroup Theory.* Oxford: Clarendon Press, 1995, X, 351 S. ISBN: 0-19-851194-9.

[5] Martin Huschenbett und Manfred Kufleitner. „Ehrenfeucht-Fraissé games on omega-terms“. In: *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014).* Hrsg. von Ernst W. Mayr und Natacha Portier. Bd. 25. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014, S. 374–385. ISBN: 978-3-939897-65-1.

[6] Neil Immerman. „Nondeterministic space is closed under complementation“. In: *SIAM Journal on computing* 17.5 (1988), S. 935–938.

[7] Manfred Kufleitner. „The Trotter-Weil Hierarchy“. Habilitation. Univeristät Stuttgart, 2013.

[8] Manfred Kufleitner und Pascal Weil. „On logical hierarchies within $\mathbf{FO}^2$-definable languages“. In: *Logical Methods in Computer Science* 8.3:11 (2012), S. 1–30.

[9] Kamal Lodaya, Paritosh K. Pandya und Simoni S. Shah. „Marking the chops: An unambiguous temporal logic“. In: *Fifth IFIP International Conference On Theoretical Computer Science – TCS 2008.* Hrsg. von Giorgio Ausiello u. a. Bd. 273. IFIP International Federation for Information Processing. Springer US, 2008, S. 461–476. ISBN: 978-0-387-09679-7.

[10] Ana Moura. „The word problem for ω-terms over $\mathbf{DA}$“. In: *Theoretical Computer Science* 412.46 (2011), S. 6556–6569.

[11] Dominique Perrin und Jean-Éric Pin. *Infinite Words. Automata, Semigroups, Logic and Games.* Bd. 141. Pure and Applied Mathematics. Academic Press, 2004. ISBN: 978-0-12-532111-2.

[12] Jean-Éric Pin. *Varieties of Formal Languages.* Aus dem Französischen übers. von A. Howie. North Oxford Academic Publishers Ltd, 1986. ISBN: 0-946536-12-0.

[13] Joseph G. Rosenstein. *Linear Orderings.* Bd. 98. Academic press, 1982.

[14] Thomas Schwentick, Denis Thérien und Heribert Vollmer. „Partially-ordered two-way automata: A new characterization of **DA**". In: *Developments in Language Theory.* Springer. 2002, S. 239–250.

[15] Róbert Szelepcsényi. „The method of forced enumeration for nondeterministic automata". In: *Acta Informatica* 26.3 (1988), S. 279–284.

[16] Denis Thérien und Thomas Wilke. „Over words, two variables are as powerful as one quantifier alternation". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing.* ACM. 1998, S. 234–240.

[17] Peter Trotter und Pascal Weil. „The lattice of pseudovarieties of idempotent semigroups and a non-regular analogue". In: *Algebra Universalis* 37.4 (1997), S. 491–526.

[18] Heribert Vollmer. *Introduction to Circuit Complexity. A Uniform Approach.* Springer, 1999. ISBN: 3-540-64310-9.

[19] Philipp Weis und Neil Immerman. „Structure theorem and strict alternation hierarchy for $\mathbf{FO}^2$ on words". In: *Logical Methods in Computer Science* 5 (2009), S. 1–23.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

(Jan Philipp Wächter)