Technische Universität Berlin **SoSe 2019**
Fakultät II, Institut für Mathematik
Sekretariat MA 6–2, Antje Schulz
Dr. Ambros Gleixner
Holger Eble, Dr. Frank Lutz, Sarah Morell

Programmierprojekt: Young Tableaux

Teil 1: Visualisierung

1.1 Definition

Ein *Young Tableau* ist eine linksbündig ausgerichtete Auflistung von Reihen beschrifteter quadratischer Kästchen, sodass folgende Bedingungen erfüllt sind:

- Die Anzahl der Kästchen pro Reihe fällt (von oben nach unten).
- In jedem Kästchen steht eine Zahl aus $\mathbb{Z}_{\geq 1}$.
- Innerhalb jeder Reihe steigen die Zahlen (von links nach rechts).
- Innerhalb jeder Spalte steigen die Zahlen strikt (von oben nach unten).

1	2	2	3	3	5
2	3	5	5		
4	4	6	6		
5	6				

Abbildung 1: Beispiel eines Young Tableaus

1.2 Aufgabenstellung

Legen Sie einen Projektordner `YoungTableau` und eine Datei `visual.py` an.
Schreiben Sie eine Funktion `print_tex`, welche die Visualisierung eines übergebenen Young Tableaus als LaTeX-Code in die Datei `visual.tex` schreibt. Dabei soll ein zusätzlicher Eingabeparameter `boxlength` berücksichtigt werden, der die Länge eines quadratischen Kästchens festlegt.
Für die Visualisierung soll das TeX-Package TikZ benutzt werden, vgl. `visual_temp.tex` auf der ISIS-Seite. Kästchen sollen wie dort exemplarisch beschrieben mit

```
\draw[...] ... rectangle ...
```

gezeichnet und mit `\node` beschriftet werden. Die letzte Reihe beginnt am Koordinatenursprung.

Teil 2: Parser

2.1 Wörter

Zu jedem Young Tableau T lässt sich durch Auflisten seiner Reihen von unten nach oben ein Wort $w(T)$ über $\mathbb{Z}_{\geq 1}$ zuordenen. Beispielsweise liefert das Young Tableau von Abbildung 1 das Wort

$$\text{"5 6 4 4 6 6 2 3 5 5 1 2 2 3 3 5"} \ .$$

2.2 Aufgabenstellung

Legen Sie eine Datei `words.py` in Ihrem Projektordner an. Schreiben Sie dort eine Funktion `parse_word(v)`, die für ein Wort v über $\mathbb{Z}_{\geq 1}$ ein Young Tableau T mit $w(T) = v$ erstellt oder einen `ValueError("no young tableau")` wirft, falls v kein Young Tableau beschreibt.

Implementieren Sie eine Funktion `parse(row, file)`, die auf die Datei `file`, in der ein Wort pro Zeile gespeichert ist, zugreift und die Zeile `row` ($\in \mathbb{Z}_{\geq 0}$) gemäß obiger Anweisung verarbeitet.

Teil 3: Algebraische Strukturen

3.1 Einfügen in Young Tableaux

Sei T ein Young Tableau und $x \in \mathbb{Z}_{\geq 1}$ ein Element, das in T eingefügt werden soll. Wir bezeichnen mit T_i die i-te Reihe von T wobei T_1 die oberste Reihe beschreibt. Das Einfügen von x in T soll folgendermaßen geschehen:

1) Setze $i = 1$.

2) Gibt es kein Element aus T_i, das echt größer als x ist, füge x am rechten Ende dieser Reihe hinzu und gebe das so entstehende Young Tableau zurück. Andernfalls nimm das linkeste aller echt größeren Elemente y aus T_i, ersetze y durch x und wiederhole Schritte 2) mit der nächsten (möglicherweise leeren) Reihe T_{i+1} und dem Element y.

Wir schreiben $(T \leftarrow x)$ für das so entstehende Young Tableau.
Zum Beispiel ergibt sich für das Tableau T aus Abbildung 1 und $x = 1$ für $(T \leftarrow x)$ das Tableau

1	1	2	3	3	5
2	2	5	5		
3	4	6	6		
4	6				
5					

.

Abbildung 2: Resultierendes Young Tableau $(T \leftarrow x)$ für $x = 1$ und T aus Abbildung 1.

3.2 Multiplikation von Young Tableaux

Seien S und T zwei Young Tableaux und $w(T) = x_1 \ldots x_s$ das Wort von T. Wir definieren das Produkt von S und T durch die binäre Operation

$$S \bullet T := ((((S \leftarrow x_1) \leftarrow x_2) \leftarrow \ldots) \leftarrow x_{s-1}) \leftarrow x_s \ .$$

Proposition 1. *Die Operation $\bullet$ auf der Menge der Young Tableaux definiert ein Monoid $\mathcal{M}_{YT}$.*

3.3 Wörter

Auch auf Wörtern über $\mathbb{Z}_{\geq 1}$ kann eine Multiplikation definiert werden. Elemente werden wie üblich durch Konkatenierung multipliziert. Es gilt etwa

$$2\ 3\ 1 \cdot 3\ 9\ 2 = 2\ 3\ 1\ 3\ 9\ 2 \ .$$

Hierdurch wird ebenfalls ein Monoid beschrieben, welches wir mit $\mathcal{M}_W$ bezeichnen. Wir geben nun eine Relation $\sim_K$ auf $\mathcal{M}_W$ an: Für zwei Wörter $w_1, w_2 \in \mathcal{M}_W$ gilt $w_1 \sim_K w_2$, falls w_2 aus w_1 durch endlich viele Operationen (auf Teilwörtern der Länge 3) der folgenden Form hervorgeht:

- $K_1 :\ *yzx* \mapsto *yxz*$, falls $x < y \leq z$
- die zu K_1 inverse Operation

- $K_2: \ *xzy* \mapsto *zxy*$, falls $x \leq y < z$
- die zu K_2 inverse Operation

Das Zeichen $*$ steht hier für eine endliche, möglicherweise leere Zeichenkette über $\mathbb{Z}_{\geq 1}$. Beispielsweise gilt 4 2 1 3 5 $\sim_K$ 2 4 3 1 5, da wir wie folgt entwickeln können:

$$4\ 2\ 1\ 3\ 5 \overset{K_1^{-1}}{\rightsquigarrow} 4\ 2\ 3\ 1\ 5 \overset{K_2^{-1}}{\rightsquigarrow} 2\ 4\ 3\ 1\ 5$$

Lemma 2. *Die Relation $\sim_K$ beschreibt eine Äquivalenzrelation auf $\mathcal{M}_W$.*

Lemma 3. *Die Menge $\overline{\mathcal{M}}_W := \mathcal{M}_W / \sim_K$ wird mit der induzierten (d.h. von $\mathcal{M}_W$ geerbten) Operation zu einem Monoid.*

Wir schreiben $\bar{w} := \pi \circ w$ mit der Quotientenabbildung $\pi\colon \mathcal{M}_W \longrightarrow \overline{\mathcal{M}}_W$. Beachten und verifizieren Sie anhand von Beispielen, dass die Äquivalenzrelation $\sim_K$ so erstellt wurde, dass sie verträglich mit der Einfügeoperation aus Teil 3.1 ist, d.h. aus der Definition erhalten wir anhand elementarer Betrachtungen folgende Aussage:

Proposition 4. *Für ein Young Tableau T und ein $x \in \mathbb{Z}_{\geq 1}$ gilt $\bar{w}(T \leftarrow x) = \bar{w}(T) \cdot \bar{w}(x)$ in $\overline{\mathcal{M}}_W$, wobei wir x auf der rechten Seite als einelementiges Young Tableau auffassen.*

Corollary 5. *Für zwei Young Tableaux S und T gilt $\bar{w}(S \bullet T) = \bar{w}(S) \cdot \bar{w}(T)$ in $\overline{\mathcal{M}}_W$.*

Lemma 6. *Die Abbildung $\bar{w}\colon \mathcal{M}_{YT} \longrightarrow \overline{\mathcal{M}}_W$ ist surjektiv.*

Beweisen Sie alle drei Lemmata. Konstruieren Sie insbesondere zu *jedem* Wort w ein Young Tableau T mit zu w equivalentem Wort $w(T)$. **Dies ist das (theoretische) Kernstück dieses Programmierprojekts!**
Tatsächlich ist $\bar{w}$ auch injektiv, was schwieriger zu zeigen ist und wir als gegeben hinnehmen.

Theorem 7. *Die Abbildung $\bar{w}\colon \mathcal{M}_{YT} \longrightarrow \overline{\mathcal{M}}_W$ ist ein Isomorphismus.*

3.4 Aufgabenstellung

Beweisen Sie die Lemmata 2, 3 und 6.
Machen Sie sich klar, was für Propositionen 1 zu zeigen ist und was davon offensichtlich ist.
Leiten Sie Korollar 5 aus Proposition 4 her.
Legen Sie eine Datei `ytmath.py` in Ihrem Projektordner an. Schreiben Sie dort eine Funktion `row_insert`, welche für ein Young Tableau T und ein $x \in \mathbb{Z}_{\geq 1}$ das Young Tableau $(T \leftarrow x)$ zurückgibt. Implementieren sie dort auch die Multiplikation (siehe 3.2) und denken Sie an das neutrale Element.
Verfahren Sie ähnlich mit der Multiplikation von Wörtern und implementieren Sie die Funktionen `K1`, `K1_inv`, `K2` und `K2_inv`, siehe unten.

Die Klassen `youngtableau` und `word`

Legen Sie eine Datei `struc.py` in Ihrem Projektordner an.

4.1 Die Klasse `youngtableau`

Implementeren Sie die Klasse `youngtableau` mit den folgenden Methoden:

- `__init__(self, word)`: Erstellt ein Young Tableau aus einem Wort, sieheTeil 2.
- `visual(self, boxlength, file)`: Speichert die Visualisierung in der Datei `file` ab.
- `row_insert(self, x)`: Gibt das Young Tableau ($\texttt{self} \leftarrow x$) zurück.
- `word(self)`: Gibt $w(\texttt{self})$ als `word` zurück.

Schreiben Sie zudem Funktionen

- `create_from(row, file)`, welche ein Young Tableau aus Zeile `row` ($\in \mathbb{Z}_{\geq 0}$) von Datei `file` zurückgibt, siehe Konvention aus Teil 2.
- `multiply(S, T)`, welche das Produkt von S und T (als `youngtableau`) zurückgibt.

4.2 Die Klasse `word`

Implementieren Sie die Klasse `word`, welche ein Wort über $\mathbb{Z}_{\geq 1}$ als Liste speichert und folgende Methoden bereitstellt:

- `K1(self, index)`: Betrachtet das Unterwort $\texttt{self}[\texttt{index}{:}\texttt{index}+3]$ der Länge 3 und gibt das durch K_1 modifizierte Wort (siehe 3.3) zurück, falls die Voraussetzungen erfüllt sind, oder wirft einen `ValueError("K-operation impossible")`.
- `K1_inv(self, index)`: analog
- `K2(self, index)`: analog
- `K2_inv(self, index)`: analog
- `youngtableau(self)`: Gibt $\bar{w}^{-1}(\overline{\texttt{self}})$ als `youngtableau` zurück.

Schreiben Sie zudem Funktionen

- `mult_classes(word1, word2)`, welche einen Repräsentant von $\overline{\texttt{word1}} \cdot \overline{\texttt{word2}}$ (Multiplikation in $\overline{\mathcal{M}}_W$) als `word` zurückgibt.
- `are_equiv(word1, word2)`, die `True` zurückgibt, falls $\texttt{word1} \sim_K \texttt{word2}$ gilt und ansonsten `False` zurückgibt.

In der Datei `struc.py` sollten überwiegend Aufrufe der in den Teilen 1 bis 3 implementierten Module erfolgen. Modifizieren oder erweitern Sie diese, falls nötig.

Abgabe

Das Projekt muss bis spätestens Donnerstag, den 4. Juli 2019, beim CoMa-Team vorgestellt werden. Vereinbaren Sie hierfür einen Termin mit Ihrer Betreuerin bzw. Ihrem Betreuer.