

# DER FOCKRAUM ALS LABYRINTH: WISSENSREPRÄSENTATION UND SPRACHVERARBEITUNG AM BEISPIEL DES *Mouse–Maze*–PROBLEMS

Peter beim Graben<sup>1</sup>, Markus Huber<sup>1</sup>, Ronald Römer<sup>1</sup>, Ingo Schmitt<sup>1</sup> und Matthias Wolff<sup>1</sup>

<sup>1</sup>BTU–Cottbus–Senftenberg  
peter.beimgraben@b-tu.de

**Kurzfassung:** Wir betrachten die syntaktische Sprachverarbeitung am Beispiel des *Mouse–Maze*–Problems, indem wir der Maus mitteilen, wo sie ein im Labyrinth verstecktes Käsestück auffinden kann. Wir definieren eine neuartige Art von Normalform für kontextfreie Grammatiken, die es erlaubt, Zustandsbeschreibungen eines aus psycholinguistischer Hinsicht besonders geeigneten *Left–Corner–Parsers* durch Terme in einer Termalgebra auszudrücken. Indem wir grammatikalische Symbole auf „Füllervektoren“ und Baumpositionen auf „Rollenvektoren“ abbilden, konstruieren wir eine Vektoriell–Symbolische Architektur, in welcher Parserzustände durch Tensorprodukt Darstellungen im Fockraum der Quantenfeldtheorie repräsentiert werden. Inkrementell und interaktiv zu verarbeitende Worte werden somit durch Operatoren in einem geometrischen Raum dargestellt.

## 1 Motivation

Einer Einschätzung des Kulturwissenschaftlers Hocke [1] gemäß, ist die Welt ein Labyrinth, in welchem manieristische Künstler, Menschen, oder, am allgemeinsten, kognitive Agenten sich zurechtfinden müssen. Dazu bedarf es mentaler Strukturen und Prozesse für Wahrnehmung, Wissensrepräsentation, Kommunikation und Problemlösung [2].

In einer Reihe von Beiträgen haben Römer et al. [3], Schmitt et al. [4] und andere diese kognitiven Leistungen an einem „Spielzeug–Modell“, dem *Mouse–Maze*–System, erforscht. Dabei wird eine virtuelle Maus in einem simulierten Labyrinth ausgesetzt, worin sie ein Stück Käse aufzufinden hat. Die Maus ist mit Sensoren ausgestattet, die ihr erlauben, ihre aktuelle Position in  $x$ – und  $y$ –Koordinaten zu bestimmen und das dortige Vorhandensein (oder Nicht–Vorhandensein) von Käse zu detektieren. Indem die Maus das Labyrinth exploriert, gelangt sie zu einer Wissensrepräsentation ihrer Umgebung in Form von semantischen Merkmal–Wert–Relationen, die sie zur Problemlösung, dem Auffinden des Käsestücks, heranziehen kann.

Eine Alternative zur Umgebungsexploration ist Kommunikation: Ein informierter Beobachter kann der Maus mitteilen: „Käse liegt im *Käsefeld*“, wobei *Käsefeld* ein geeignet gewählter Bezeichner für ein Koordinatenpaar  $(x_K, y_K)$  ist, auf dem sich das Käsestück befindet. Die Aufgabe der Sprachverarbeitung besteht dann darin, aus der linearen Zeichenkette eine hierarchische Syntaxstruktur aufzubauen, die dann in eine semantische Merkmal–Wert–Relation überführt werden kann.

In diesem Beitrag wollen wir uns auf das Problem der syntaktischen Sprachverarbeitung, des *Parsing*, beschränken und zeigen, wie ein interaktiver und inkrementeller [5] *Left–Corner–Parser* (LC–Parser) algebraisch und geometrisch charakterisiert werden kann. Dazu konstruieren wir eine kontextfreie Grammatik für die Mitteilung, deren Syntax wir durch eine Termalgebra ausdrücken [6, 7]. Inkrementell zu verarbeitende Worte werden dann als Operatoren auf dieser Termalgebra dargestellt. Schließlich konstruieren wir eine *Vektoriell–Symbolische Architektur* (VSA) [8, 9], die Terme auf Vektoren in einem geometrischen Raum, dem sogenannten

*Fockraum* der Quantenfeldtheorie abbildet [6, 10], so dass wir eine Matrixdarstellung der Wort-Operatoren auf diesem Raum erhalten. Wir beweisen, dass die symbolischen Termtransformationen homomorph auf diese Matrixoperatoren abgebildet werden.

## 2 Syntaktische Sprachverarbeitung

Wir unterteilen die Konstruktion einer Sprachverarbeitungs-VSA in drei Schritte: 1. die symbolische Beschreibung durch kontextfreie Grammatiken und interaktiv-inkrementelle LC-Parser, 2. die algebraische Beschreibung durch Termalgebren und Termtransformationen, 3. die geometrische Beschreibung durch Tensorproduktdarstellungen und Operatoren im Fockraum.

### 2.1 Symbolische Beschreibung

Um den Satz „Käse liegt im *Käsefeld*“ verstehen und interpretieren zu können, muss die virtuelle Maus seinen Phrasenstrukturbaum rekonstruieren, den wir in Abb. 1 angeben.

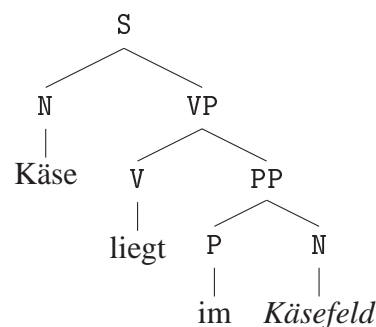


Abbildung 1 – Phrasenstrukturbaum der Maus-Mitteilung

In Abb. 1 steht S für „Satz“, das Startsymbol der Grammatik, N für ein Substantiv (Nomen), VP für eine Verbalphrase (Prädikat), V für Verb, PP für eine Präpositionalphrase (adverbiale Bestimmung des Ortes) und schließlich P für Präposition.

Indem wir jede Verzweigung des Phrasenstrukturbaums Abb. 1 als Produktionsregel interpretieren, erhalten wir eine kontextfreie Grammatik von Phrasenstrukturregeln in der sogenannten Chomsky-Normalform [7, 11],

$$S \rightarrow N \quad VP \quad (1a)$$

$$VP \rightarrow V \quad PP \quad (1b)$$

$$PP \rightarrow P \quad N \quad (1c)$$

$$N \rightarrow \text{Käse} \quad (1d)$$

$$N \rightarrow \text{Käsefeld} \quad (1e)$$

$$V \rightarrow \text{liegt} \quad (1f)$$

$$P \rightarrow \text{im}, \quad (1g)$$

wobei wir die Regeln (1a — 1c) als *syntaktische Regeln* und Regeln (1d — 1g) als *lexikalische Regeln* bezeichnen. Abstrakt wird eine kontextfreie Grammatik als Quadrupel  $G = (\mathbf{T}, \mathbf{N}, S, R)$  definiert, so dass in unserem Beispiel  $\mathbf{T} = \{\text{Käse}, \text{Käsefeld}, \text{liegt}, \text{im}\}$  als Menge der Terminalsymbole (Worte),  $\mathbf{N} = \{S, N, VP, V, PP, P\}$  als Menge der Nicht-Terminalsymbole (Kategorien),  $S \in \mathbf{N}$  als distinguiertes Startsymbol und  $R \subset \mathbf{N} \times (\mathbf{N} \cup \mathbf{T})^*$  als Menge der Regeln angegeben

werden. Eine Regel  $r = (A, \gamma) \in R$  wird oft als Produktion  $r : A \rightarrow \gamma$  geschrieben, wobei  $A \in \mathbf{N}$  eine Kategorie und  $\gamma \in (\mathbf{N} \cup \mathbf{T})^*$  eine endliche Zeichenkette aus Terminalen und/oder Kategorien der Länge  $n = |\gamma|$  darstellt.

Kontextfreie Grammatiken können durch Kellerautomaten prozessiert werden [11]. In psycholinguistischer Hinsicht sind dafür insbesondere LC-Parser von Interesse, bei denen eingabegetriebene *Bottom-Up*- und erwartunggetriebene *Top-Down*-Prozesse einander abwechseln [12]. Im Folgenden werden wir anstelle der kompletten Automaten eine vereinfachte funktionale Beschreibung wählen, bei denen ein interaktiv eingelesenes Wort einen temporären Phrasenstrukturbaum inkrementell auf seinen LC-Nachfolger abbildet (Abb. 2).

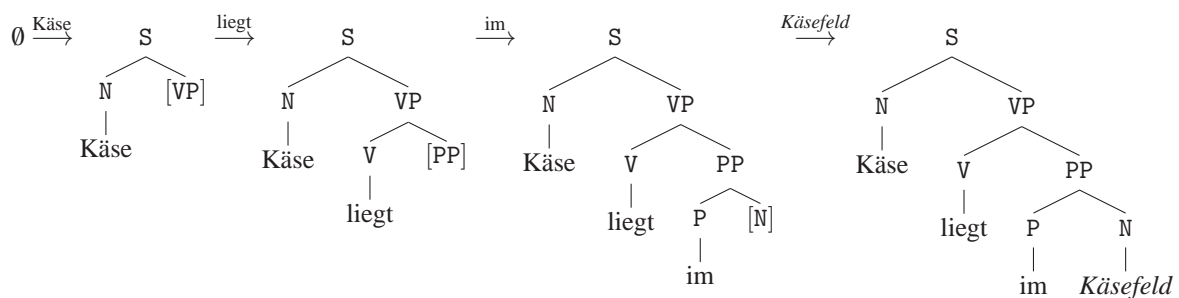


Abbildung 2 – Interaktiver LC-Parser der Maus-Mitteilung

Der Parse in Abb. 2 beginnt mit einem „Vakuumzustand“, dem leeren Baum  $\emptyset$ . Das Einlesen des ersten Wortes „Käse“, das nach der lexikalischen Regel (1d) der Grammatik  $G$  als Substantiv  $N$  kategorisiert wird, führt zur Vorhersage einer Verbalphrase  $VP$ , gemäß Regel (1a), wobei  $N$  als *left corner* („linke Ecke“) die Vorhersage auslöst. Wir folgen Hale [12], indem wir vorhergesagte Kategorien in eckige Klammern setzen:  $[VP]$  bedeutet vorhergesagte Verbalphrase. Im zweiten Schritt bestätigt das eingelesene Verb (V) „liegt“ die vorhergesagte  $VP$  und löst gleichzeitig als linke Ecke die Vorhersage einer Präpositionalphrase  $[PP]$  nach Regel (1b) aus. In dieser Weise wird schrittweise der Phrasenstrukturbaum Abb. 1 aufgebaut.

## 2.2 Algebraische Beschreibung

Im nächsten Schritt wollen wir die temporären Bäume der LC-Analyse Abb. 2 algebraisch ausdrücken. Dazu bietet sich eine Formalisierung durch Termalgebren an [6, 7]. Eine Termalgebra ist durch eine *Signatur*, eine Funktion, die Symbolen eine ganzzahlige Arität zuordnet, gekennzeichnet. Dadurch werden die Symbole als Funktionsbezeichner interpretiert, deren Argumente Terme sind. Um diesen Begriff auf kontextfreie Grammatiken anwenden zu können, müssen die Kategorien eine eindeutig bestimmte Arität aufweisen. Zu diesem Zweck definieren wir eine neuartige Normalform: Eine kontextfreie Grammatik  $G = (\mathbf{T}, \mathbf{N}, S, R)$  liege in *Term-Normalform* vor, wenn für jede Kategorie  $A \in \mathbf{N}$  gilt: Falls  $A$  für  $n \in \mathbb{N}$  in den Regeln  $r_1 : A \rightarrow \gamma_1$  bis  $r_n : A \rightarrow \gamma_n$  expandiert wird, folgt  $|\gamma_1| = \dots = |\gamma_n|$ .<sup>1</sup>

Wie man leicht sieht, erfüllen die Regeln (1a — 1g) unserer Maus-Grammatik  $G$  diese Bedingung. Daher können wir eine Termalgebra definieren, indem die Menge der Variablen

<sup>1</sup>Es bleibt noch zu zeigen, dass zu jeder kontextfreien Grammatik eine Term-Normalform existiert. Es ist einsichtig, dass jede kontextfreie Grammatik, die in Chomsky-Normalform vorliegt, in Term-Normalform überführbar ist. Dazu müssen eventuell nur zusätzliche Nicht-Terminalsymbole eingeführt werden, um die rechten Seiten von Produktionsregeln auf die gleiche Länge aufzufüllen. Für jedes dieser zusätzlichen Symbole muss noch eine Regel eingeführt werden, die dieses Symbol auf das leere Wort abbildet. Eine Konstruktion für allgemeine kontextfreie Grammatiken steht noch aus.

$\mathbf{V} = \mathbf{N} \cup \mathbf{T}$  als Domäne einer Signatur  $\text{rank} : \mathbf{V} \rightarrow \mathbb{N}_0$  angegeben wird, so dass 1.  $\text{rank}(a) = 0$  für alle  $a \in \mathbf{T}$ , d.h. Terminale sind nulläre Funktionen, also Konstanten. 2.  $\text{rank}(A) = |\gamma|$  für Kategorien  $A \in \mathbf{N}$ , die durch Regeln  $A \rightarrow \gamma$  expandiert werden, wobei  $|\gamma|$  für alle Regeln, die  $A$  expandieren, in der Term–Normalform eindeutig bestimmt ist.

Falls die Grammatik  $G$  zudem in Chomsky–Normalform vorliegt, gilt insbesondere für alle Kategorien  $A \in \mathbf{N}$ , die ausschließlich in lexikalischen Regeln vorkommen,  $\text{rank}(A) = 1$ , d.h. lexikalische Kategorien ( $\mathbf{N}, \mathbf{V}, \mathbf{P}$ ) sind unäre Funktionen. Während  $\text{rank}(A) = 2$  für alle Kategorien  $A \in \mathbf{N}$ , die ausschließlich in syntaktischen Regeln erscheinen, d.h. syntaktische Kategorien sind binäre Funktionen.

Für eine allgemeine kontextfreie Grammatik  $G$  in Term–Normalform definieren wir die Termalgebra  $\mathfrak{T}(G)$  induktiv: 1. jedes Terminalsymbol  $a \in \mathbf{T}$  ist ein Term:  $a \in \mathfrak{T}(G)$ . Sei  $A \in \mathbf{N}$  eine Kategorie mit  $\text{rank}(A) = k$  und seien  $t_0, \dots, t_{k-1} \in \mathfrak{T}(G)$  Terme, dann ist  $A(t_0, \dots, t_{k-1}) \in \mathfrak{T}(G)$  ein Term.

Um zusätzlich auch LC–Bäume beschreiben zu können, erweitern wir die Signatur um die vorhergesagten Kategorien  $\mathbf{P} = \{[\text{VP}], [\text{PP}], [\text{N}]\}$ , die wir als Konstanten mit  $\text{rank}(P) = 0$  für  $P \in \mathbf{P}$  interpretieren. Die erweiterte Termalgebra sei dann  $\mathfrak{T}_{\text{LC}}(G)$ .

In dieser LC–Termalgebra wird z.B. der Baum aus Schritt 2 des LC–Parses in Abb. 2 durch den Term  $S(\mathbf{N}(\text{Käse})[\text{VP}])$  kodiert, während  $S(\mathbf{N}(\text{Käse})\text{VP}(\mathbf{V}(\text{liegt})[\text{PP}]))$  den nachfolgenden Baum ergibt. Es liegt daher nahe, die strukturellen Operationen des LC–Parsers durch Termtransformationen zu beschreiben. Dazu definieren wir die folgenden (partiellen) Funktionen auf  $\mathfrak{T}_{\text{LC}}(G)$ .

$$\text{cat}(A(t_0, \dots, t_k)) = A \quad (2a)$$

$$\text{ex}_i(A(t_0, \dots, t_k)) = t_i \quad (2b)$$

$$\text{cons}_k(A, t_0, \dots, t_k) = A(t_0, \dots, t_k), \quad (2c)$$

dabei liefert die Funktion  $\text{cat} : \mathfrak{T}_{\text{LC}}(G) \rightarrow \mathbf{N}$  die Kategorie, d.h. das Funktionssymbol  $A$  des Terms  $A(t_0, \dots, t_k)$ . Die Funktionen  $\text{ex}_i : \mathfrak{T}_{\text{LC}}(G) \rightarrow \mathfrak{T}_{\text{LC}}(G)$  für Termextraktion und  $\text{cons}_k : \mathbf{N} \times \mathfrak{T}_{\text{LC}}(G)^{k+1} \rightarrow \mathfrak{T}_{\text{LC}}(G)$  als Termkonstruktor sind lediglich partiell vorgegeben, wobei  $A(t_0, \dots, t_k) \in \text{Dom}(\text{ex}_i)$ , falls  $k = \text{rank}(A) - 1$  und  $i < k$ , sowie  $(A, t_0, \dots, t_k) \in \text{Dom}(\text{cons}_k)$ , wenn  $k = \text{rank}(A) - 1$ .

Mithilfe der Termtransformationen (2a — 2c) können wir die interaktive und inkrementelle Wirkung eines eingelesenen Wortes  $a \in \mathbf{T}$  durch einen Term–Operator  $[[a]] : \mathfrak{T}_{\text{LC}}(G) \rightarrow \mathfrak{T}_{\text{LC}}(G)$  ausdrücken. Für den Übergang von LC–Baum 2 zu 3 in Abb. 2 ergibt sich so

$$\begin{aligned} [[\text{liegt}]](S(\mathbf{N}(\text{Käse})[\text{VP}])) = \\ \text{cons}_1(\text{cat}(S(\mathbf{N}(\text{Käse})[\text{VP}]), \text{ex}_0(S(\mathbf{N}(\text{Käse})[\text{VP}])), \text{VP}(\mathbf{V}(\text{liegt})[\text{PP}])) = \\ S(\mathbf{N}(\text{Käse})\text{VP}(\mathbf{V}(\text{liegt})[\text{PP}])). \quad (3) \end{aligned}$$

## 2.3 Geometrische Beschreibung

Im dritten Schritt konstruieren wir einen Homomorphismus  $\psi : \mathfrak{T}_{\text{LC}}(G) \cup \mathbf{N} \rightarrow \mathcal{F}$ , der Terme aus  $\mathfrak{T}_{\text{LC}}(G)$  und Kategorien aus  $\mathbf{N}$  auf einen Vektorraum  $\mathcal{F}$  abbildet, so dass die Transformationen (2a — 2c) strukturerhaltend auf Vektorraum–Operatoren, d.h. auf Matrizen abgebildet werden.

Auch hier gehen wir wieder induktiv vor, indem wir als erstes die Symbole in  $\mathfrak{T}_{\text{LC}}(G) \cup \mathbf{N}$  betrachten. Ihre Anzahl ist die Mächtigkeit  $n = |\mathbf{T} \cup \mathbf{N} \cup \mathbf{P}|$ . Für jedes Symbol  $s_i \in \mathbf{T} \cup \mathbf{N} \cup \mathbf{P}$  definieren wir einen Füllvektor  $\vec{f}_i = \psi(s_i) \in \mathcal{F}$  und nennen den Unterraum  $\mathcal{V}_F = \{\sum_i \lambda_i \vec{f}_i : \lambda_i \in$

$\mathbb{R}, 1 \leq i \leq n\}$  Füllerraum. Sei weiterhin  $m = \max(\{|\gamma| : (A \rightarrow \gamma) \in R\})$  die Länge der längsten Produktion in der Grammatik  $G$ . Dann definieren wir  $m + 1$  sogenannte *Rollenvektoren*  $\vec{r}_i$ , die den *Rollenraum*  $\mathcal{V}_R = \{\sum_i \lambda_i \vec{r}_i : \lambda_i \in \mathbb{R}, 0 \leq i \leq m\}$  aufspannen.<sup>2</sup> Dabei kodieren  $\vec{r}_0$  den 1. Tochterknoten,  $\vec{r}_1$  den 2. Tochterknoten, usw. bis zum letzten Tochterknoten  $\vec{r}_{m-1}$ . Die verbleibende Rolle  $\vec{r}_m$  bindet den Mutterknoten im Phrasenstrukturbaum der Grammatik  $G$ .

Sei schließlich  $A(t_0, \dots, t_k) \in \mathfrak{T}_{\text{LC}}(G)$  ein Term, dann heie

$$\psi(A(t_0, \dots, t_k)) = \psi(A) \otimes \vec{r}_m \oplus \psi(t_0) \otimes \vec{r}_0 \oplus \dots \oplus \psi(t_k) \otimes \vec{r}_k \quad (4)$$

die *Tensorproduktarstellung* des Terms  $A(t_0, \dots, t_k) \in \mathfrak{T}_{\text{LC}}(G)$  in  $\mathcal{F}$ . Man kann sich leicht davon berzeugen, dass prinzipiell unendliche Rekursion von Termen zu einem unendlich-dimensionalen Darstellungsraum

$$\mathcal{F} = \bigoplus_{p=0}^{\infty} \left( \mathcal{V}_F \otimes \bigotimes_{q=1}^p \mathcal{V}_R \right) \oplus \mathcal{V}_R, \quad (5)$$

dem Fockraum der Quantenfeldtheorie fhrt [6, 10], der zur Beschreibung von fluktuierenden Vielteilchenzuständen unentbehrlich ist: Jeder  $p$ -dimensionale Unterraum („Sektor“) in der direkten Summe (5) entspricht dabei einem Quantensystem von  $p$  Teilchen. In unserem Beispiel gibt  $p$  die Hierarchie-Ebene im Phrasenstrukturbaum Abb. 1 an, whrend  $q$  die Pfade vom Wurzelknoten zu den Blttern indiziert.

In der Quantenfeldtheorie ist ein Zustand  $|0\rangle \neq 0$  als Vakuumzustand ausgezeichnet. Dieser beschreibt ein System mit  $p = 0$  Teilchen. Setzen wir dies in (5) ein, ergibt sich gerade der Unterraum der Rollenvektoren  $\mathcal{V}_R$ . Wir knnen daher den leeren Term durch einen beliebigen Rollenvektor darstellen; eine geeignete Wahl ist die „Mutterrolle“,  $\psi(\emptyset) = \vec{r}_m$ .

Mit der Tensorproduktarstellung (4) lassen sich die Bilder der obigen Terme berechnen. Fr den 2. LC-Parse-Schritt aus Abb. 2 erhalten wir z.B.

$$\begin{aligned} \psi(\text{S}(\text{N}(\text{Kse})[\text{VP}])) &= \psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}(\text{Kse})) \otimes \vec{r}_0 \oplus \psi([\text{VP}]) \otimes \vec{r}_1 = \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus (\psi(\text{N}) \otimes \vec{r}_2 \oplus \psi(\text{Kse}) \otimes \vec{r}_0) \otimes \vec{r}_0 \oplus \psi([\text{VP}]) \otimes \vec{r}_1 = \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}) \otimes \vec{r}_2 \otimes \vec{r}_0 \oplus \psi(\text{Kse}) \otimes \vec{r}_0 \otimes \vec{r}_0 \oplus \psi([\text{VP}]) \otimes \vec{r}_1. \end{aligned} \quad (6)$$

Dementsprechend ergibt sich fr Schritt 3

$$\begin{aligned} \psi(\text{S}(\text{N}(\text{Kse})\text{VP}(\text{V}(\text{liegt})[\text{PP}]))) &= \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}(\text{Kse})) \otimes \vec{r}_0 \oplus \psi(\text{VP}(\text{V}(\text{liegt})[\text{PP}])) \otimes \vec{r}_1 = \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus (\psi(\text{N}) \otimes \vec{r}_2 \oplus \psi(\text{Kse}) \otimes \vec{r}_0) \otimes \vec{r}_0 \oplus \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus (\psi(\text{N}) \otimes \vec{r}_2 \oplus \psi(\text{Kse}) \otimes \vec{r}_0) \otimes \vec{r}_0 \oplus \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus (\psi(\text{N}) \otimes \vec{r}_2 \oplus \psi(\text{Kse}) \otimes \vec{r}_0) \otimes \vec{r}_0 \oplus \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus (\psi(\text{N}) \otimes \vec{r}_2 \oplus \psi(\text{Kse}) \otimes \vec{r}_0) \otimes \vec{r}_0 \oplus \psi([\text{PP}]) \otimes \vec{r}_1 \otimes \vec{r}_1 = \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}) \otimes \vec{r}_2 \otimes \vec{r}_0 \oplus \psi(\text{Kse}) \otimes \vec{r}_0 \otimes \vec{r}_0 \oplus \\ &= \psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}) \otimes \vec{r}_2 \otimes \vec{r}_0 \oplus \psi(\text{Kse}) \otimes \vec{r}_0 \otimes \vec{r}_0 \oplus \psi([\text{PP}]) \otimes \vec{r}_1 \otimes \vec{r}_1. \end{aligned} \quad (7)$$

Im Fockraum wird dann die interaktive und inkrementelle Wirkung eines eingelesenen Wortes  $a \in \mathbf{T}$  durch einen Matrix-Operator  $[[a]]_\psi : \mathcal{F} \rightarrow \mathcal{F}$  dargestellt. Fr den bergang von Schritt 2 zu Schritt 3 im LC-Parse Abb. 2 erhalten wir mithin

<sup>2</sup> Wir nehmen hier an, dass der Koeffizientenkrper von  $\mathcal{F}$  gleich  $\mathbb{R}$  sei; im Rahmen der Quantenkognition wrde man eher  $\mathbb{C}$  whlen.

$$\begin{aligned}
[[\text{liegt}]]_\psi(\psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}) \otimes \vec{r}_2 \otimes \vec{r}_0 \oplus \psi(\text{Käse}) \otimes \vec{r}_0 \otimes \vec{r}_0 \oplus \psi([\text{VP}]) \otimes \vec{r}_1) = \\
\psi(\text{S}) \otimes \vec{r}_2 \oplus \psi(\text{N}) \otimes \vec{r}_2 \otimes \vec{r}_0 \oplus \psi(\text{Käse}) \otimes \vec{r}_0 \otimes \vec{r}_0 \oplus \\
\oplus \psi(\text{VP}) \otimes \vec{r}_2 \otimes \vec{r}_1 \oplus \psi(\text{V}) \otimes \vec{r}_2 \otimes \vec{r}_0 \otimes \vec{r}_1 \oplus \psi(\text{liegt}) \otimes \vec{r}_0 \otimes \vec{r}_0 \otimes \vec{r}_1 \oplus \psi([\text{PP}]) \otimes \vec{r}_1 \otimes \vec{r}_1. \quad (8)
\end{aligned}$$

Um zeigen zu können, dass die Funktion  $\psi$  ein Homomorphismus ist, definieren wir die folgenden linearen Abbildungen auf  $\mathcal{F}$ .

$$\mathbf{cat}(\vec{u}) = (\text{id} \otimes \vec{r}_m^+)(\vec{u}) \quad (9a)$$

$$\mathbf{ex}_i(\vec{u}) = (\text{id} \otimes \vec{r}_i^+)(\vec{u}) \quad (9b)$$

$$\mathbf{cons}_k(\vec{a}, \vec{u}_0, \dots, \vec{u}_k) = \vec{a} \otimes \vec{r}_m \oplus \vec{u}_0 \otimes \vec{r}_0 \oplus \dots \oplus \vec{u}_k \otimes \vec{r}_k, \quad (9c)$$

dabei bezeichnet  $\text{id}$  die identische Funktion. Die Linearformen  $\vec{r}_i^+$  aus dem Dualraum  $\mathcal{V}_R^*$  sind adjungiert zu den Rollenvektoren  $\vec{r}_i$ , so dass  $\vec{r}_i^+(\vec{r}_j) = \delta_{ij}$  mit den Kronecker-Symbolen  $\delta_{ij} = 0(1)$  für  $i \neq j(i = j)$ .<sup>3</sup>

Aus den Definitionen (9a — 9c) folgt das Darstellungstheorem: Die Tensorprodukt Darstellung  $\psi: \mathfrak{T}_{\text{LC}}(G) \cup \mathbf{N} \rightarrow \mathcal{F}$  ist ein Homomorphismus hinsichtlich der Termtransformationen (2a — 2c). Es gilt:

$$\mathbf{cat}(\psi(A(t_0, \dots, t_k))) = \psi(\mathbf{cat}(A(t_0, \dots, t_k))) \quad (10a)$$

$$\mathbf{ex}_i(\psi(A(t_0, \dots, t_k))) = \psi(\mathbf{ex}_i(A(t_0, \dots, t_k))) \quad (10b)$$

$$\mathbf{cons}_k(\psi(A), \psi(t_0), \dots, \psi(t_k)) = \psi(\mathbf{cons}_k(A, t_0, \dots, t_k)). \quad (10c)$$

### 3 Beweis des Darstellungstheorems

Der Beweis des obigen Theorems ergibt sich durch einfache Rechnung.

$$\begin{aligned}
\mathbf{cat}(\psi(A(t_0, \dots, t_k))) &= (\text{id} \otimes \vec{r}_m^+)(\psi(A(t_0, \dots, t_k))) = \\
&(\text{id} \otimes \vec{r}_m^+)(\psi(A) \otimes \vec{r}_m \oplus \psi(t_0) \otimes \vec{r}_0 \oplus \dots \oplus \psi(t_k) \otimes \vec{r}_k) = \psi(A) = \psi(\mathbf{cat}(A(t_0, \dots, t_k))),
\end{aligned}$$

$$\begin{aligned}
\mathbf{ex}_i(\psi(A(t_0, \dots, t_k))) &= (\text{id} \otimes \vec{r}_i^+)(\psi(A(t_0, \dots, t_k))) = \\
&(\text{id} \otimes \vec{r}_i^+)(\psi(A) \otimes \vec{r}_m \oplus \psi(t_0) \otimes \vec{r}_0 \oplus \dots \oplus \psi(t_k) \otimes \vec{r}_k) = \psi(t_i) = \psi(\mathbf{ex}_i(A(t_0, \dots, t_k))),
\end{aligned}$$

$$\begin{aligned}
\mathbf{cons}_k(\psi(A), \psi(t_0), \dots, \psi(t_k)) &= \psi(A) \otimes \vec{r}_m \oplus \psi(t_0) \otimes \vec{r}_0 \oplus \dots \oplus \psi(t_k) \otimes \vec{r}_k = \\
&\psi(\mathbf{cons}_k(A, t_0, \dots, t_k)) \quad \square
\end{aligned}$$

---

<sup>3</sup> In einem Hilbertraum können die  $\vec{r}_i^+$  durch ein Biorthonormalsystem ausgedrückt werden:  $\langle \vec{r}_i^+ | \vec{r}_j \rangle = \delta_{ij}$ .

## 4 Diskussion und Ausblick

Syntaktische Sprachverarbeitung ist ein wichtiger erster Schritt zum semantischen Sprachverständnis und zur Wissensrepräsentation. Am Beispiel des *Mouse–Maze*–Problems diskutieren wir die Syntexanalyse einer Mitteilung „Käse liegt im *Käsefeld*“, durch welche wir einer intelligenten Maus helfen, ihren Weg durch ein Labyrinth zu finden. Wir beschreiben die Mitteilung durch eine kontextfreie Grammatik und ihre Verarbeitung durch einen psycholinguistisch plausiblen *Left–Corner–Parser*. Zustandsbeschreibungen des Parsers geben wir als geordnete Phrasenstrukturbäume an, die wir durch Terme einer Termalgebra ausdrücken können, wenn die Grammatik eine neuartige Form von Normalform besitzt. Durch Tensorprodukt Darstellungen werden Terme auf Vektoren im Fockraum abgebildet, der in der Quantenfeldtheorie der Vielteilchensysteme verwendet wird.

Unser Beitrag zeigt, dass komplexe symbolische Datenstrukturen, wie z.B. Phrasenstrukturbäume der Syntexanalyse, geometrisch in Vektoriell–Symbolischen Architekturen darstellbar sind. Darüber hinaus können interaktive Kommunikationssysteme, die sprachliche Eingaben inkrementell zu verarbeiten haben, geometrisch realisiert werden, indem Eingaben als Matrix–Operatoren auf vektoriell repräsentierten Zuständen dargestellt werden.

Das übergeordnete Ziel besteht darin, dass wir den Übergang von der syntaktischen Ebene in die semantische Ebene unter Berücksichtigung der Parsing–Strukturen realisieren wollen. Diese Ebene dient der *Repräsentation* von konkreten Objekten, Eigenschaften und Beziehungen, welche in der realen Welt der Fall sein können und von einem „abstrakten“ Weltmodell erfasst werden. Zu diesem Zweck müssen die natürlichsprachlichen Äußerungen in eine Metasprache überführt werden, welche jedoch nicht so ausdrucksmächtig wie die natürliche Sprache sein muss. Was in der Metasprache ausgedrückt wird, entspricht dann einer Situationsbeschreibung oder einem Handlungszusammenhang (zwischen einer aktuellen Situation und einer Folgesituation) im „Weltmodell“. Dazu muss aus dem natürlichen Satz, die für das Weltmodell relevante Information extrahiert werden, d.h. die betreffenden Objekte und deren Beziehungen müssen formal repräsentiert werden. Im vorliegenden Beitrag haben wir gezeigt, dass die Terminalsymbole eines Satzes durch die Tensorverknüpfung von  $p$ –Teilchensystemen repräsentiert werden. Um Objekte und Beziehungen unterscheiden zu können, müssen wir auch die Konstituenten des Satzes im Fockraum identifizieren können, so dass wir den betreffenden Unterräumen die entsprechende Bedeutung zuweisen können. Die Lösung dieser Problemstellung wird Gegenstand zukünftiger Arbeiten sein.

Vektoriell–Symbolischen Architekturen lassen sich einerseits durch neuronale Netzwerke implementieren [5, 9]. Andererseits deutet ihre enge Verbindung mit der Quantentheorie auf neuartige Anwendungen von Methoden der Quantenkognition hin [10, 6, 4].

## Literatur

- [1] HOCKE, G. R.: *Die Welt als Labyrinth. Manier und Manie in der europäischen Kunst*. Rowohlt, Hamburg, 1957.
- [2] ANDERSON, J. R.: *Kognitive Psychologie*. Spektrum, Heidelberg, 1996.
- [3] RÖMER, R., M. HUBER, und G. WIRSCHING: *Ein Beitrag zur Gedankengeometrie kognitiver Systeme*. In O. JOKISCH (Hrsg.), *Tagungsband der 27. Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, Bd. 81 d. Reihe *Studentexte zur Sprachkommunikation*, S. 101 – 110. TU Dresden Press, Dresden, 2016.
- [4] SCHMITT, I., R. RÖMER, G. WIRSCHING, und M. WOLFF: *Semantische Repräsentation*

*strukturierter Objekte im Hilbertraum und deren Verarbeitung nach quantenmechanischen Formalismen.* In J. TROUVAIN, I. STEINER, und B. MÖBIUS (Hrsg.), *Tagungsband der 28. Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, Bd. 86 d. Reihe *Studientexte zur Sprachkommunikation*, S. 270 – 277. TU Dresden Press, Dresden, 2017.

- [5] BEIM GRABEN, P., S. GERTH, und S. VASISHTH: *Towards dynamical system models of language-related brain potentials.* *Cognitive Neurodynamics*, 2(3), S. 229 – 255, 2008.
- [6] BEIM GRABEN, P. und S. GERTH: *Geometric representations for minimalist grammars.* *Journal of Logic, Language and Information*, 21(4), S. 393 – 432, 2012.
- [7] KRACHT, M.: *The Mathematics of Language.* Mouton de Gruyter, Berlin, 2003.
- [8] GAYLER, R. W.: *Vector symbolic architectures are a viable alternative for Jackendoff's challenges.* *Behavioral and Brain Sciences*, 29, S. 78 – 79, 2006.
- [9] SMOLENSKY, P.: *Tensor product variable binding and the representation of symbolic structures in connectionist systems.* *Artificial Intelligence*, 46(1-2), S. 159 – 216, 1990.
- [10] AERTS, D.: *Quantum structure in cognition.* *Journal of Mathematical Psychology*, 53(5), S. 314 – 348, 2009.
- [11] HOPCROFT, J. E. und J. D. ULLMAN: *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie.* Addison–Wesley, Bonn, 1994.
- [12] HALE, J. T.: *What a rational parser would do.* *Cognitive Science*, 35(3), S. 399 – 443, 2011.