

MASTERARBEIT

*Mathematisch-logische Aspekte von Beweisrepräsentationsstrukturen*

Angefertigt am  
Mathematischen Institut

Vorgelegt der  
Mathematisch-Naturwissenschaftlichen Fakultät der  
Rheinischen Friedrich-Wilhelms-Universität Bonn

Mai 2009

Von

Marcos Cramer

Aus

Buenos Aires, Argentinien



---

# Danksagung

---

Ich möchte mich an dieser Stelle bei allen Personen bedanken, die mir bei der Erstellung dieser Masterarbeit geholfen haben. Ohne sie wäre diese Arbeit nicht möglich gewesen. Mein herzlicher Dank gilt PETER KOEPKE für die Betreuung meiner Arbeit und DANIEL KÜHLWEIN für die vielen fachlichen sowie stilistischen Kommentare. Außerdem bedanke ich mich bei BERNHARD SCHRÖDER und JIP VELDMAN für ihre nützlichen Bemerkungen bei der Entwicklung der PRS-Semantik.

Besonderer Dank geht natürlich auch an meine Frau und an meine Eltern für ihre allumfassende Unterstützung.

## **Eidesstattliche Erklärung**

Hiermit erkläre ich, Marcos Cramer, an Eides statt, dass ich die Masterarbeit, „Mathematisch-logische Aspekte von Beweisrepräsentationsstrukturen“, selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Bonn, den 4. Mai 2009

Unterschrift

---

# Inhaltsverzeichnis

---

<b>Inhaltsverzeichnis</b>	<b>i</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Die Naproche-Sprache und die mathematische Formelsprache</b>	<b>5</b>
2.1 Syntax der Formelsprache . . . . .	5
2.2 Semantik der Formelsprache . . . . .	7
2.3 Naproche-Sprache . . . . .	8
<b>3 Beweisrepräsentationsstrukturen</b>	<b>13</b>
3.1 PRS-Syntax . . . . .	13
3.2 PRS-Semantik . . . . .	19
<b>4 Das Formelbild einer PRS</b>	<b>25</b>
4.1 Lambda-Kalkül . . . . .	26
4.2 Definition des Formelbildes . . . . .	26
4.3 Beispiel einer Formelbild-Konstruktion . . . . .	30
4.4 Wohldefiniertheit des Formelbildes . . . . .	35
<b>5 Äquivalenz einer PRS und ihres Formelbildes</b>	<b>49</b>
5.1 Hilfskonstruktionen . . . . .	49
5.2 Äquivalenzbeweis . . . . .	51
<b>6 Ausblick</b>	<b>65</b>
<b>7 Fazit</b>	<b>69</b>
<b>A Die Grammatik der Naproche-Sprache</b>	<b>71</b>
<b>Literaturverzeichnis</b>	<b>79</b>



# Kapitel 1

---

## Einleitung

---

Die Sprache mathematischer Texte, wie sie in wissenschaftlichen Journalen und mathematischen Lehrbüchern zu finden sind, kombiniert natürliche Sprache mit mathematischen Zeichen. Obwohl in mathematischen Texten Elemente aus formalen Sprachen übernommen werden, und der Gebrauch von mehrdeutigen Ausdrücken vermieden wird, handelt es sich bei dieser Sprache keinesfalls um eine formale Sprache: Fast die gesamte Vielfalt natürlichsprachlicher Ausdrucksweisen findet man auch in mathematischen Texten. Dadurch werden manchmal auch Ausdrücke genutzt, die syntaktisch ambig sind, und deren beabsichtigte Bedeutung nur durch Verständnis des Kontextes bestimmt werden kann.

Andererseits werden bestimmte Ausdrücke in mathematischen Texten auf sehr systematische Weise immer wieder gebraucht. Durch explizite Definition erhalten viele Ausdrücke eine eindeutige Bedeutung. Außerdem werden die soeben erwähnten syntaktischen Ambiguitäten von Mathematikern auch gerne vollständig vermieden. Deshalb lässt sich annehmen, dass mathematische Texte mit den Methoden der Computerlinguistik leichter zu analysieren sein sollten als nicht-mathematische Texte.

Ein Hauptbestandteil mathematischer Texte sind mathematische Beweise. Doch was ist ein mathematischer Beweis? Anders als bei nicht-mathematischen Beweisen (z.B. in anderen Wissenschaften oder vor Gericht), darf bei einem mathematischen Beweis nicht induktiv, sondern nur deduktiv geschlossen werden. Das heißt, dass nicht von Einzelfällen auf Allgemeines geschlossen werden kann, sondern nur solche Schlussfolgerungen möglich sind, die als logische Notwendigkeiten angesehen werden.

Ausgehend von GOTTLOB FREGES *Begriffsschrift* [4] hat sich um die Jahrhundertwende der Begriff des *formalen Beweises* gebildet. Dabei handelt es sich um einen Beweis in einer formalen Sprache, bei dem nur rein mechanische syntaktische Regeln angewandt werden und somit jeder logische Schritt des Beweises

explizit ist. BERTRAND RUSSELL und ALFRED NORTH WHITEHEAD haben in der *Principia Mathematica* [11] formale Beweise für Sätze aus den Grundlagen der Mathematik verfasst. Allerdings wurde durch dieses Werk auch ersichtlich, dass formale Beweise für den alltäglichen Gebrauch in der Mathematik nicht praktikabel sind und sich daher nicht als Standardschreibweise mathematischer Beweise durchsetzen können. Es blieb allerdings die Einsicht vorherrschend, dass sich mathematische Beweise im Prinzip immer in formale Beweise umwandeln lassen.

Mit dem Aufkommen von Computern ist es möglich geworden, das Formalisieren von Beweisen für Mathematiker angenehmer zu machen. Als Beispiel sei hier das *Mizar*-System [10] aufgeführt: Dieses System besteht aus einer formalen Sprache, der *Mizar*-Sprache, und einem Beweisprüfer, der Beweise in dieser Sprache auf logische Korrektheit überprüft. Der Beweisprüfer kann eine Folge einfacher logischer Umformungen selber durchführen, so dass der Mathematiker nicht mehr jeden einzelnen logischen Schritt im formalen Beweis aufschreiben muss. Die *Mizar*-Sprache ist zwar eine formale Sprache, orientiert sich jedoch an der mathematischen Fachsprache. Das heißt, dass sie auch bestimmte natürlichsprachliche Ausdrücke, die häufig in mathematischen Texten genutzt werden, mit eindeutigen syntaktischen Anwendungsregeln und eindeutiger Bedeutung einbezieht. Dies macht die *Mizar*-Sprache für Mathematiker einfacher zu lesen als manch andere formale Sprache. Trotzdem weicht die Sprache von *Mizar* noch stark von der gewöhnlichen Sprache mathematischer Texte ab.

Das von PETER KOEPKE (Mathematik, Universität Bonn) und BERNHARD SCHRÖDER (Linguistik, Universität Duisburg-Essen) geleitete Projekt *Naproche* hat unter anderem zum Ziel, eine formale Sprache für mathematisches Schreiben zu entwickeln, die der natürlichen Sprache mathematischer Texte sehr nahe kommt. Mathematische Texte, die in dieser Sprache verfasst sind, sollen einerseits für einen Mathematiker leicht lesbar sein und wie natürliche mathematische Texte wirken, andererseits aber von einem Computer-Programm (dem *Naproche-System*) in ein übliches prädikatenlogisches Format übertragbar und auf ihre Korrektheit überprüfbar sein. Diese Sprache, auch in ihrer derzeitigen prototypischen Version, wird als *Naproche-Sprache* bezeichnet. Einen in dieser Sprache verfassten Text nennen wir *Naproche-Text*.

Die Umwandlung von *Naproche*-Texten in Prädikatenlogik erfolgt über ein linguistisches Zwischen-Format, das an die in der Computerlinguistik mittlerweile als Standardmethode geltenden *Diskursrepräsentationstheorie* angelehnt ist. Dieses Zwischen-Format nennen wir *Beweisrepräsentationsstruktur* (engl. *Proof Representation Structure, PRS*).

Zum Überprüfen von PRSen macht sich *Naproche* automatische Beweissysteme zu Nutzen: Somit kann man in *Naproche* einen Beweis schreiben, ohne jeden logischen Schritt aufzuschreiben. Die einzelnen Schritte müssen nur so klein sein,



dass der verwendete automatische Beweiser sie bewältigen kann. Ein so aufgeschriebener Beweis kann von einem Computer in einen formalen Beweis, in dem jeder logische Schritt explizit ist, umgewandelt werden.

In der vorliegenden Arbeit beziehen wir uns auf zwei Versionen des Naproche-Systems: Im Juli 2008 wurde Naproche 0.1 als erste Version, die mit PRSen arbeitet, fertig gestellt. Nachdem Anfang 2009 die Naproche-Sprache stark erweitert wurde, wird diese zurzeit (April 2009) in Naproche 0.2 implementiert.

Wie die PRSen in Naproche 0.1 aus einem Naproche-Text erstellt und auf Korrektheit überprüft werden, wurde schon von in der Magisterarbeit von NICKOLAY KOLEV [8] und in der Diplomarbeit von DANIEL KÜHLWEIN [9] ausgeführt. In der hier vorliegenden Arbeit sollen hingegen mathematisch-logische Aspekte der Beweisrepräsentationsstrukturen behandelt werden, die in diesen Arbeiten noch nicht detailliert behandelt wurden.

Kapitel 2 stellt die Sprache von Naproche 0.2 und die in sie eingebettete mathematische Formelsprache vor. In Kapitel 3 definieren wir, was PRSen sind, und stellen erstmals eine formale, modelltheoretische Semantik für PRSen vor, die an üblichen Semantiken der Diskursrepräsentationstheorie angelehnt ist. Durch diese Semantik haben wir ein formales Verständnis davon, was eine PRS aussagt.

Kapitel 4 definiert das *Formelbild*, eine Übersetzung von PRSen in Formeln der Prädikatenlogik erster Stufe. Damit eine PRS ein Formelbild hat, muss sie einige Bedingungen erfüllen. PRSen, die diese Bedingungen erfüllen, bezeichnen wir als *wohlgeformte* PRSen. In Kapitel 5 zeigen wir, dass jede wohlgeformte PRS zu ihrem Formelbild äquivalent ist, so dass man das Formelbild als eine korrekte Übersetzung bezeichnen kann. Für den dafür gebrauchten Äquivalenzbegriff ist die in Kapitel 3 definierte PRS-Semantik unabdingbar.

Kapitel 6 gibt einen Ausblick auf weitere theoretische Aspekte, die im Rahmen des Naproche-Projekts noch zu behandeln sind.



## Kapitel 2

---

# Die Naproche-Sprache und die mathematische Formelsprache

---

Ein mathematischer Text enthält sowohl natürlichsprachliche Konstruktionen als auch mathematische Symbole. Die natürlichsprachlichen Konstruktionen können mit linguistischen Methoden analysiert werden. Die mathematischen Symbole hingegen treten gewöhnlicherweise in der Form von *Termen* und *Formeln* einer prädikatenlogischen Sprache auf, die Zeichen für mathematische Funktionen und Relationen enthält. Diese Sprache bezeichnen wir als *Formelsprache*.

### 2.1 Syntax der Formelsprache

Die hier vorgestellte Syntax der Formelsprache ist eine theoretische Version der Formelsprache, die im jetzigen Naproche-System implementiert ist.

**Definition 2.1.1.** *Das Alphabet der Formelsprache*

Zum Alphabet der Formelsprache zählen die folgenden Zeichen:

1. Variablenzeichen:  $k, l, m, n, u, v, w, x, y, z, x_0, x_1, \dots, v_0, v_1, \dots$
2. Junktoren:  $\perp, \top, \neg, \vee, \wedge, \rightarrow, \leftrightarrow$
3. Quantoren:  $\forall, \exists$
4. Klammern:  $(, )$
5. Relationszeichen: Für jedes  $n \in \mathbb{N}$  eine eventuell leere Menge von  $n$ -stelligen Relationszeichen, wobei für  $n = 2$  das Symbol „ $\doteq$ “ in der Menge sein muss.
6. Konstantenzeichen: Eine eventuell leere Menge von Konstantenzeichen

7. Funktionszeichen: Für jedes  $n \in \mathbb{N}$  eine eventuell leere Menge von  $n$ -stelligen Funktionszeichen

Die in Punkten 1-4 aufgezählten Zeichen nennen wir auch *logische Zeichen*. Anstelle von „ $\doteq$ “ kann man auch das gewöhnlichere Gleichheitszeichen „ $=$ “ schreiben. Wir bevorzugen aber in dieser Arbeit meist „ $\doteq$ “, da wir „ $=$ “ auch benutzen, wenn wir *über* die Formelsprache oder die Naproche-Sprache reden.

**Bemerkung.** Hier wird nicht nur eine einzige Formelsprache definiert, sondern eine Formelsprache für jede mögliche Wahl von nicht-logischen Zeichen. Für den Rest dieser Arbeit sei  $L$  eine Formelsprache.

**Definition 2.1.2.** *Terme einer Formelsprache*

Die Menge der  $L$ -Terme wird definiert als die minimale Menge, für die folgendes gilt:

- Jedes Variablenzeichen und jedes Konstantenzeichen ist ein  $L$ -Term.
- Wenn  $t_1, \dots, t_n$   $L$ -Terme sind und  $f$  ein  $n$ -stelliges Funktionszeichen ist, so ist  $f(t_1, \dots, t_n)$  ein  $L$ -Term.
- Wenn  $t_1, t_2$   $L$ -Terme sind und  $f$  ein zweistelliges Funktionszeichen ist, so sind  $t_1ft_2$  und  $(t_1ft_2)$   $L$ -Terme.

**Definition 2.1.3.** *Formeln einer Formelsprache*

Die Menge der  $L$ -Formeln wird definiert als die minimale Menge, für die folgendes gilt:

1.  $\perp$  und  $\top$  sind  $L$ -Formeln.
2. Wenn  $t_1, \dots, t_n$   $L$ -Terme sind und  $R$  ein  $n$ -stelliges Relationszeichen ist, so ist  $R(t_1, \dots, t_n)$  eine  $L$ -Formel.
3. Wenn  $t_1, t_2$   $L$ -Terme sind und  $R$  ein zweistelliges Relationszeichen ist, so sind  $t_1Rt_2$  und  $(t_1Rt_2)$   $L$ -Formeln.
4. Wenn  $\varphi$  und  $\psi$   $L$ -Formeln sind, so sind  $\neg\varphi$ ,  $(\varphi \wedge \psi)$ ,  $(\varphi \vee \psi)$ ,  $(\varphi \rightarrow \psi)$  und  $(\varphi \leftrightarrow \psi)$   $L$ -Formeln.
5. Wenn  $\varphi$  eine  $L$ -Formel und  $x$  ein Variablenzeichen ist, so sind  $\forall x\varphi$  und  $\exists x\varphi$   $L$ -Formeln.

Die Klammern, die in Punkt 4 vorkommen, können auch weggelassen werden. Die Interpretation dieser Formeln ist so, als ob die Klammern so spät wie möglich geschlossen werden.  $\varphi \wedge \psi \rightarrow \chi$  ist also eine Abkürzung für  $(\varphi \wedge (\psi \rightarrow \chi))$ .

**Definition 2.1.4.** Sei  $L$  eine Formelsprache. Dann ist  $F_L$  die Menge aller Formeln von  $L$ , und  $A_L$  die Menge aller Konstanten-, Funktions- und Relationszeichen von  $L$ .

Freie und gebundene Variablen, sowie die Substitutionsfunktion  $\varphi \frac{t}{x}$ , werden wie gewöhnlich definiert (detaillierte Definitionen finden sich z.B. in EBBINGHAUS et al. [3], S. 24, 55-56). Mit  $\text{frei}(\varphi)$  bezeichnen wir die Menge aller freien Variablen einer Formel  $\varphi$ . Bei einem Term  $t$  bezeichnet  $\text{frei}(t)$  die Menge der Variablen in  $t$ . Als *L-Satz* bezeichnen wir eine *L-Formel*, die keine freien Variablen enthält.

## 2.2 Semantik der Formelsprache

### Definition 2.2.1. *L-Struktur*

Eine *L-Struktur*  $\mathfrak{A}$  mit Universum  $U$  ist ein Paar  $\langle U, \beta \rangle$ , wobei  $U$  eine Menge ist und  $\beta$  eine Abbildung ist, die jedem Konstantenzeichen von  $L$  ein Element in  $U$ , jedem  $n$ -stelligem Funktionszeichen von  $L$  ein Funktion von  $U^n$  nach  $U$  und jedem  $n$ -stelligem Relationszeichen von  $L$  eine Teilmenge von  $U^n$  zuordnet. Für ein Konstantenzeichen  $c$ , ein Funktionszeichen  $f$  und ein Relationszeichen  $R$  schreiben wir anstatt  $\beta(c)$ ,  $\beta(f)$  bzw.  $\beta(R)$  auch  $\mathfrak{A}(c)$ ,  $\mathfrak{A}(f)$  bzw.  $\mathfrak{A}(R)$ . Wir bezeichnen das Universum von  $\mathfrak{A}$  auch mit  $|\mathfrak{A}|$ .

### Definition 2.2.2. *Teilbelegung*

Sei  $U$  eine Menge. Eine *U-Teilbelegung* ist eine partielle Funktion von der Menge der Variablenzeichen von  $L$  nach  $U$ .

### Definition 2.2.3. *Erweiterung einer Teilbelegung auf L-Terme*

Sei  $\sigma$  eine *U-Teilbelegung*. Die *Erweiterung von  $\sigma$  auf L-Terme*,  $\bar{\sigma}$ , wird rekursiv definiert:

1.  $\bar{\sigma}(v) := \sigma(v)$  für jedes Variablenzeichen  $v$  von  $L$ .
2.  $\bar{\sigma}(c) := \mathfrak{A}(c)$  für jedes Konstantenzeichen  $c$  von  $L$ .
3.  $\bar{\sigma}(f(x_1, \dots, x_n)) := \mathfrak{A}(f)(\bar{\sigma}(x_1), \dots, \bar{\sigma}(x_n))$  für jedes  $n$ -stellige Funktionszeichen  $f$  von  $L$ .

Man beachte, dass  $\bar{\sigma}$  genau für jene *L-Terme* definiert ist, die nur Variablen, auf denen  $\sigma$  definiert ist, enthalten.

### Definition 2.2.4. *Semantik der Formelsprache*

Sei  $\mathfrak{A}$  eine *L-Struktur* mit Universum  $U$  und  $\sigma$  eine *U-Teilbelegung*. Seien  $t_1, \dots, t_n$  *L-Terme* und  $\varphi, \psi$  *L-Formeln*. Wir definieren:

- $\mathfrak{A}, \sigma \models \top$  gilt in jedem Falle.
- $\mathfrak{A}, \sigma \models \perp$  gilt in keinem Falle.
- $\mathfrak{A}, \sigma \models t_1 \doteq t_2$  gdw  $\bar{\sigma}(t_1) = \bar{\sigma}(t_2)$ .
- $\mathfrak{A}, \sigma \models R(t_1, \dots, t_n)$  gdw  $\langle \bar{\sigma}(t_1), \dots, \bar{\sigma}(t_n) \rangle \in \mathfrak{A}(R)$ .

- $\mathfrak{A}, \sigma \models \neg\varphi$  gdw  $\mathfrak{A}, \sigma \models \varphi$  nicht gilt.
- $\mathfrak{A}, \sigma \models (\varphi \rightarrow \psi)$  gdw entweder  $\mathfrak{A}, \sigma \models \varphi$  nicht gilt oder  $\mathfrak{A}, \sigma \models \psi$  gilt
- $\mathfrak{A}, \sigma \models \exists x\varphi$  gdw es ein Element  $a \in U$  gibt, so dass  $\mathfrak{A}, \sigma_x^a \models \varphi$ , wobei
 
$$\sigma_x^a(v) := \begin{cases} \sigma(v) & \text{wenn } v \neq x. \\ a & \text{wenn } v = x. \end{cases}$$

Für einen  $L$ -Satz  $\chi$  definieren wir

- $\mathfrak{A} \models \chi$  genau dann, wenn  $\mathfrak{A}, \emptyset \models \chi$ .

## 2.3 Naproche-Sprache

Die Naproche-Sprache ist eine formale Sprache, in der man mathematische Texte schreiben kann, die wie natürlichsprachliche mathematischen Texte aussehen, und auch wie diese interpretiert werden. Zurzeit gibt es nur eine Englisch-sprachige Version der Naproche-Sprache. Diese orientiert sich also an der mathematischen Fachsprache Englisch-sprachiger Texten. Als Beispiel dieser Fachsprache sei hier der Beweis für die Irrationalität von  $\sqrt{2}$  aus der Einführung in die Zahlentheorie von HARDY und WRIGHT [6] aufgeführt:

If  $\sqrt{2}$  is rational, then the equation  $a^2 = 2b^2$  is soluble in integers  $a$ ,  $b$  with  $(a, b) = 1$ . Hence  $a^2$  is even, and therefore  $a$  is even. If  $a = 2c$ , then  $4c^2 = 2b^2$ ,  $2c^2 = b^2$ , and  $b$  is also even, contrary to the hypothesis that  $(a, b) = 1$ .

An diesem Beispiel sehen wir, dass die mathematische Fachsprache die Syntax und Semantik der Gemeinsprache verwendet, und somit auch ihre Komplexität und einen Teil ihrer Ambiguitäten übernimmt. Allerdings unterscheiden sich mathematische Texte in mehrerer Hinsicht von gemeinsprachlichen Texten:

- Natürlichsprachliche Ausdrücke werden mit mathematischen Symbolen und Formeln, die syntaktisch wie Nominalphrasen und Teilsätze fungieren, verbunden.
- Mehrdeutige Konstruktionen, deren beabsichtigte Bedeutung sich nicht leicht bestimmen lässt, werden im Allgemeinen gemieden.
- Mathematische Symbole machen den Text eindeutiger, z.B. durch den Gebrauch von Variablen anstatt anaphorischen Pronomen.
- Annahmen können eingeführt und zurückgezogen werden. Der obige Beweis zum Beispiel ist ein Widerspruchsbeweis: Am Anfang wird angenommen,

dass  $\sqrt{2}$  rational ist. Die darauf folgenden Aussagen werden als Schlussfolgerungen aus dieser Annahme angesehen, nicht als absolute Wahrheiten. Schließlich führt die Annahme zu einem Widerspruch. Sie wird dann zurückgezogen, und es wird geschlossen, dass  $\sqrt{2}$  irrational ist. Wir bezeichnen den Teil des Textes von der Einführung bis zum Zurückziehen einer Annahme als *Skopus* der Annahme.

- Mathematische Texte sind sehr strukturiert. Global werden sie in Komponenten wie Definitionen, Lemmata, Sätze und Beweise eingeteilt. Innerhalb eines Beweises werden Annahmen geschachtelt, so dass die Skopi der Annahmen eine hierarchische Beweisstruktur definieren.
- Definitionen fügen neue Zeichen und Ausdrücke zum Vokabular hinzu und legen ihre Bedeutung fest.
- Beweisschritte werden häufig durch Referenz auf Ergebnisse aus anderen Texten oder früheren Stellen im selben Text begründet.

In der Naproche-Sprache ist Mehrdeutigkeit nicht möglich. Die anderen der soeben erwähnten Eigenschaften der mathematischen Fachsprache wurden in die Naproche-Sprache übernommen.

Die Grammatik der Sprache von Naproche 0.2, geschrieben als Definite-Klausel-Grammatik in Prolog mit GULP-Merkmalen [2], findet sich im Anhang A. Die Grammatik bezieht sich auf das Modul `fo_grammar`, das die Formelsprache definiert. Für unsere theoretischen Überlegungen kann man die Definitionen der Formelsprache in Abschnitt 2.1 anstelle dieses Moduls übernehmen.

Diese Grammatik definiert sowohl, nach dem klassischen Verständnis von „Grammatik“, die möglichen Sätze, als auch die möglichen Aneinanderreihung von Sätzen in Naproche-Texten. Das erste bezeichnen wir auch als *Mikrostruktur*, das zweite als *Makrostruktur*.

Der folgende Naproche-Text, der eine Einführung in die Gruppentheorie darstellt, verdeutlicht vor allem die Möglichkeiten der Makrostruktur:\*

Axiom 1: For all  $x, y, z$ ,  $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ .

Axiom 2: For all  $x$ ,  $e \cdot x = x$  and  $x \cdot e = x$ .

Axiom 3: For all  $x$ ,  $x \cdot x^{-1} = e$  and  $x^{-1} \cdot x = e$ .

Lemma: If  $u \cdot x = x$  then  $u = e$ .

---

\*Die Schreibweise  $x^{-1}$  für die Umkehrfunktion ist in der obigen Definition der Formelsprache akzeptabel, wenn man dies als verschönerte Schreibweise für  $x\hat{-}1$  betrachtet, und  $-1$  als Konstantenzeichen ansieht. Dann ist  $\hat{-}$  ein zweistelliges Funktionszeichen, dessen einzige Funktion es ist, mit  $-1$  als zweitem Argument die Umkehrfunktion zu repräsentieren.

Proof: Suppose that  $u \cdot x = x$ . Then  $(u \cdot x) \cdot x^{-1} = x \cdot x^{-1}$ . By axiom 1,  $u \cdot (x \cdot x^{-1}) = x \cdot x^{-1}$ . So by axiom 3  $u \cdot e = e$ . Then  $u = e$  by axiom 2. Qed.

Lemma 1: If  $x \cdot y = e$  then  $y = x^{-1}$ .

Proof: Assume  $x \cdot y = e$ . Then  $x^{-1} \cdot (x \cdot y) = x^{-1} \cdot e$ , i.e.  $(x^{-1} \cdot x) \cdot y = x^{-1}$ . Hence  $e \cdot y = x^{-1}$ , i.e.  $y = x^{-1}$ . Qed.

Theorem 1:  $(x \cdot y)^{-1} = y^{-1} \cdot x^{-1}$ .

Proof: Let  $u = (x \cdot y) \cdot (y^{-1} \cdot x^{-1})$ . Then  $u = x \cdot ((y \cdot y^{-1}) \cdot x^{-1})$  by axiom 1. So  $u = x \cdot (e \cdot x^{-1})$ , i.e.  $u = x \cdot x^{-1}$ , i.e.  $u = e$ . Thus  $(x \cdot y) \cdot (y^{-1} \cdot x^{-1}) = e$ . Hence  $(y^{-1} \cdot x^{-1}) = (x \cdot y)^{-1}$  by lemma 1. Qed.

Wie man sieht, ermöglicht die Naproche-Sprache den Gebrauch von Textstrukturierenden Ausdrücken wie „axiom“, „lemma“, „theorem“, „proof“ und „qed“ wie sie in mathematischen Texten üblich sind. Mit Ausdrücken wie „suppose that“ und „assume“ werden Annahmen eingeführt. Ein Aspekt, der die Naproche-Sprache trotz ihres natürlichsprachlichen Scheines auf gewisse Weise unnatürlich macht, ist die Tatsache, dass Annahmen nur durch ein „qed“ oder ein „thus“ geschlossen werden können. In gewöhnlichen mathematischen Texten wird der Skopus von Annahmen nicht durch rein syntaktische Textmerkmale bestimmt, sondern semantisch aus dem Text-Inhalt inferiert.

Der folgende zahlentheoretische Naproche-Text soll deutlich machen, welche Art von Mikrostrukturen möglich sind:

Define  $n$  to be square if and only if there is an  $m$  such that  $n = m^2$ .

Theorem: No square number is prime.

Proof: Let  $n$  be a square number.

Case 1:  $n = 0$  or  $n = 1$ . Then  $n$  is not prime.

Case 2:  $n \neq 0$  and  $n \neq 1$ . Then there is an  $m$  such that  $n = m^2$ . Then  $m \neq 1$  and  $m \neq n$ .  $m$  divides  $n$ , i.e.  $n$  is not prime.

So in both cases  $n$  is not prime. Qed.

Wir sehen hier also, dass Nomen („number“), Adjektive („prime“, „square“) und Verben („divides“, „is“) wie in der natürlichen Sprache verwendet werden können. Außerdem gibt es Determinative („no“) und natürlichsprachliche Quantoren („there is“). Zu beachten ist auch die Möglichkeit von Nebensätzen mit „such that“, die in mathematischen Texten häufig aufzufinden sind. Es gibt allerdings auch noch viele Einschränkungen, z.B. gibt es keinen Plural.

Die Semantik der Naproche-Sprache wird über die Übersetzung von Naproche-Texten in Beweisrepräsentationsstrukturen (PRSen) definiert. Die Übersetzung wurde in Naproche 0.2 schon implementiert. Der Übersetzungsalgorithmus für



Naproche 0.1 wurde schon detaillierter in KOLEV [8] und KÜHLWEIN [9] beschrieben. In Naproche 0.2 hat sich an dem Übersetzungsalgorithmus der schon vorher existierenden Ausdrücke nichts geändert. Hinzugekommen ist vor allem die PRS-Erstellung für Sätze, die Nomen, Verben und gegebenenfalls Adjektive enthalten; der Übersetzungsalgorithmus für diese Sätze entspricht einem Standard-Algorithmus zur Erstellung von Diskursrepräsentationsstrukturen (DRS), wie er z.B. von BLACKBURN und BOS [1] beschrieben und in Attempo [5] implementiert wurde.

Damit die Übersetzung in PRSen als Semantik für die Naproche-Sprache gelten kann, muss allerdings auch eine Semantik für PRSen geben. Diese war bisher nicht definiert worden, und ist das Thema des folgenden Kapitels.



## Kapitel 3

---

# Beweisrepräsentationsstrukturen

---

Um den Inhalt von mathematischen Texten darzustellen, nutzt Naproche ein logisch-linguistisches Format, das an Diskursrepräsentationsstrukturen [7] angelehnt ist, allerdings so abgewandelt ist, dass die im letzten Kapitel erwähnten Besonderheiten von mathematischen Texten gut dargestellt werden können. Dieses Format nennen wir Beweisrepräsentationsstrukturen (engl. *Proof Representation Structures*, PRS).

Da wir in diesem und in späteren Kapiteln häufiger mit endlichen Folgen (ab jetzt einfach *Folgen* genannt) arbeiten, seien hier einige Notationen für die Handhabung von Folgen eingeführt:

**Definition 3.0.1.** Seien  $\Sigma_1 = \langle a_1, \dots, a_n \rangle$  und  $\Sigma_2 = \langle b_1, \dots, b_m \rangle$  Folgen.

1.  $\Sigma_1(k)$  bezeichnet das  $k$ -te Element von  $\Sigma_1$ , also  $a_k$ .
2.  $a \in \Sigma_1$  bedeutet, dass  $a$  in  $\Sigma_1$  vorkommt, also dass  $a = a_i$  für ein  $i$  mit  $1 \leq i \leq n$ .
3.  $Länge(\Sigma_1)$  bezeichnet die Anzahl der Elemente in  $\Sigma_1$ , also  $n$ .
4.  $\Sigma_1 + \Sigma_2$  sei die Folge  $\langle a_1, \dots, a_n, b_1, \dots, b_m \rangle$ .

### 3.1 PRS-Syntax

Analog zu der Definition von Diskursrepräsentationsstrukturen werden PRSen und PRS-Bedingungen simultan rekursiv definiert. Die aktuelle Definition von PRSen ist der in KÜHLWEIN [9] und KOLEV [8] ähnlich. Allerdings kommt eine zusätzliche Art von PRS-Bedingungen hinzu, die „Prädikatorbedingungen“. Diese repräsentieren natürlichsprachliche Wörter (Substantive, Adjektive oder

Verben), die Prädikate ausdrücken. Solche natürlichsprachlichen Wörter nennen wir *Prädikatoren*.

**Definition 3.1.1.** *Beweisrepräsentationsstruktur (PRS)*

Gegeben ist eine Formelsprache  $L$  und eine Menge  $\Lambda$  von Prädikatoren. Eine *Beweisrepräsentationsstruktur (PRS)*  $B$  ist ein Quintupel  $\langle id_B, D_B, M_B, \Sigma_B, R_B \rangle$  mit den folgenden Eigenschaften:

- $id_B$  ist ein PRS-Identifikator.
- $D_B$  ist eine Menge von natürlichen Zahlen (die als Diskursreferenten fungieren).
- $M_B$  ist eine Menge von Termen und Formeln einer Formelsprache (die wir „mathematische Referenten“ nennen).
- $\Sigma_B$  ist eine endliche Folge von PRS-Bedingungen.
- $R_B$  ist eine Menge von PRS-Identifikatoren (die als Referenzen fungieren).

**Darstellung.** Eine PRS  $B$  mit  $id_B = i$ ,  $D_B = \{d_1, \dots, d_m\}$ ,  $M_B = \{m_1, \dots, m_n\}$ ,  $\Sigma_B = \langle c_1, \dots, c_l \rangle$  und  $R_B = \{r_1, \dots, r_k\}$  wird gewöhnlicherweise wie folgt dargestellt:

$$i$$

$d_1, \dots, d_m$	$m_1, \dots, m_n$
$c_1$	
$\vdots$	
$c_l$	
$r_1, \dots, r_k$	

In dem Rest dieser Arbeit spielen die PRS-Identifikatoren und die Referenzen keine Rolle mehr, so dass wir eine vereinfachte Darstellung benutzen:

$d_1, \dots, d_m$	$m_1, \dots, m_n$
$c_1$	
$\vdots$	
$c_l$	

**Definition 3.1.2.** *PRS-Bedingungen*

Seien  $B$  und  $C$  PRS-en,  $d, d_1, \dots, d_n$  Diskursreferenten,  $m$  ein mathematischer Referent,  $p$  ein  $n$ -stelliger Prädikator und  $f$  ein Funktionszeichen.

1.  $math\_id(d, m)$  ist eine PRS-Bedingung.

2.  $p(d_1, \dots, d_n)$  ist eine PRS-Bedingung.
3.  $holds(d)$  ist eine PRS-Bedingung.
4.  $contradiction$  ist eine PRS-Bedingung.
5.  $\neg B$  ist eine PRS-Bedingung.
6.  $B \rightarrow C$  ist eine PRS-Bedingung.
7.  $B \leftarrow C$  ist eine PRS-Bedingung.
8.  $B \leftrightarrow C$  ist eine PRS-Bedingung.
9.  $B \Rightarrow C$  ist ein PRS-Bedingung.
10.  $B := C$  ist eine PRS-Bedingung, wenn  $B$  und  $C$  die unten aufgeführten Bedingungen erfüllen.
11.  $f : B \rightarrow C$  ist eine PRS-Bedingung.
12.  $B$  ist eine PRS-Bedingung.

Punkt 10 wird syntaktisch eingeschränkt, so dass jede Definitionsbedingung genau ein neues Zeichen definiert, und nur die dafür benötigten Diskursreferenten und mathematischen Referenten einführt:

Eine Definitionsbedingung  $B := C$  darf zwei mögliche Formen annehmen:

- Die Liste der Bedingungen von  $B$  hat die Form

$$\langle \mathit{math\_id}(d_1, v_1), \dots, \mathit{math\_id}(d_k, v_k), \mathit{math\_id}(d, R(v_1, \dots, v_k)), holds(d) \rangle$$

für ein Relationszeichen  $R$  der Stelligkeit  $k$  und für Variablenzeichen  $v_1, \dots, v_k$ , und  $R$  kommt nicht in  $C$  vor. In diesem Fall sagen wir, dass  $B := C \quad R(v_1, \dots, v_k)$  mittels der Diskursreferenten  $d_1, \dots, d_k, d$  definiert, bzw. einfach dass  $B := C \quad R$  definiert.

- Die Liste der Bedingungen von  $B$  besteht aus 0 bis  $m$  verschiedenen Bedingungen der Form  $\mathit{math\_id}(d_i, v_i)$  für  $1 \leq i \leq m$  gefolgt von einer Bedingung der Form  $p(d_1, \dots, d_m)$  für einen Prädikator der Stelligkeit  $m$ , und  $p$  kommt nicht in  $C$  vor.

$$\text{Sei } \alpha_i := \begin{cases} v_i & \text{wenn } B \text{ eine Bedingung der Form } \mathit{math\_id}(d_i, v_i) \text{ enthält} \\ d_i & \text{wenn } B \text{ keine Bedingung der Form } \mathit{math\_id}(d_i, v_i) \text{ enthält} \end{cases}$$

für  $1 \leq i \leq m$

Seien  $v_{i_1}, \dots, v_{i_n}$  die Variablen der Form  $v_i$ , die in  $\{\alpha_1, \dots, \alpha_m\}$  vorkommen.

In diesem Fall sagen wir, dass  $B := C \quad p(d_1, \dots, d_m)$  mittels  $\alpha_1 \dots \alpha_m$  und mit Variablen  $v_{i_1}, \dots, v_{i_n}$  definiert, bzw. einfach dass  $B := C \quad p$  definiert.

Die PRS-Bedingungen der Form „ $p(d_1, \dots, d_n)$ “, „ $B \leftarrow C$ “, „ $B \leftrightarrow C$ “ und „ $f : B \rightarrow C$ “ wurden erst in Naproche 0.2 eingeführt. Von diesen neuartigen PRS-Bedingungen behandelt diese Arbeit nur „ $p(d_1, \dots, d_n)$ “, da diese Bedingung bei der Behandlung der PRS-Semantik von besonderem Interesse ist.\* „ $B \leftarrow C$ “, „ $B \leftrightarrow C$ “ und „ $f : B \rightarrow C$ “ werden in dieser Arbeit nicht weiter erörtert. Der Junktor „ $\rightarrow$ “ hat dieselbe Semantik wie der Junktor „ $\Rightarrow$ “ und wird daher auch nicht gesondert behandelt. (Während „ $\rightarrow$ “ aus Konditionalsätzen und All-Quantifizierungen hervorgeht, geht „ $\Rightarrow$ “ aus Annahmen und Axiomen hervor; zurzeit werden diese beiden Junktoren programmiertechnisch ein bisschen anders behandelt, obwohl sie in der Theorie gleich behandelt werden.)

**Definition 3.1.3.** *Sub-PRS*

Sei  $\Theta$  eine PRS oder eine PRS-Bedingung. Die Sub-PRSen von  $\Theta$  werden rekursiv definiert:

- Ist  $\Theta$  eine PRS, so ist  $\Theta$  eine Sub-PRS von  $\Theta$ .
- Ist  $\Theta$  eine PRS-Bedingung der Form  $\neg B$ , so ist  $B$  eine Sub-PRS von  $\Theta$ .
- Ist  $\Theta$  eine PRS-Bedingung der Form  $B \Rightarrow C$  oder  $B := C$ , so sind  $B$  und  $C$  Sub-PRSen von  $\Theta$ .
- Ist  $B$  eine Sub-PRS von  $\Theta$ , und  $C \in \Sigma_B$  eine PRS, so ist  $C$  eine Sub-PRS von  $\Theta$ .
- Ist  $B$  eine Sub-PRS von  $\Theta$ , und  $C$  eine PRS, so dass  $\neg C \in \Sigma_B$ , so ist  $C$  eine Sub-PRS von  $\Theta$ .
- Ist  $B$  eine Sub-PRS von  $\Theta$ , und sind  $C$  und  $D$  PRSen, so dass  $B \Rightarrow C \in \Sigma_B$  oder  $B := C \in \Sigma_B$ , so sind  $C$  und  $D$  Sub-PRSen von  $\Theta$ .

Eine PRS  $B$  wird als Sub-PRS einer Folge  $\Theta$  von PRS-Bedingungen bezeichnet, wenn sie Sub-PRS einer PRS-Bedingung in  $\Theta$  ist.

**Definition 3.1.4.** *Zugänglichkeit*

Anders als in *Discourse Representation Theory* ist Zugänglichkeit bei PRSen nicht nur eine Beziehung zwischen PRSen, sondern auch zwischen PRS-Bedingungen, und zwischen PRSen und PRS-Bedingungen. Wie in *Discourse Representation Theory* kann man Zugänglichkeit durch eine Unterordnungs-Beziehung definieren:

---

\*Nur durch Bedingungen der Form „ $p(d_1, \dots, d_n)$ “ ist es möglich, Diskursreferenten zu benutzen, ohne sie durch *math.id*-Bedingungen an Terme oder Formeln der Formelsprache zu binden.

Seien  $\Theta_1$  und  $\Theta_2$  PRSen oder PRS-Bedingungen.<sup>†</sup>  $\Theta_2$  ist  $\Theta_1$  *direkt untergeordnet*, wenn

- $\Theta_1$  eine PRS ist, und  $\Theta_2$  die erste Bedingung von  $\Theta_1$  ist, oder
- $\Theta_1$  und  $\Theta_2$  PRS-Bedingungen einer PRS  $B$  sind, so dass  $\Theta_1$  in der Reihenfolge der PRS-Bedingungen von  $B$  direkt vor  $\Theta_2$  kommt, oder
- $\Theta_1$  eine PRS-Bedingung der Form  $\neg\Theta_2$  ist, oder
- $\Theta_1$  eine Bedingung der Form  $\Theta_2 \rightarrow B$  ist, oder
- $\Theta_1$  eine Bedingung der Form  $\Theta_2 := B$  ist, oder
- $\Theta_1 \Rightarrow \Theta_2$  eine PRS-Bedingung ist, oder
- $\Theta_1 := \Theta_2$  eine PRS-Bedingung ist.

Die Beziehung „ $\Theta_1$  ist von  $\Theta_2$  aus zugänglich“ ist die transitive Hülle der Beziehung „ $\Theta_2$  ist  $\Theta_1$  direkt untergeordnet“.

Einen Diskursreferenten  $d$  bzw. einen mathematischen Referenten  $m$  nennen wir von  $\Theta_2$  aus zugänglich, wenn es eine von  $\Theta_2$  aus zugängliche PRS  $B$  gibt, so dass  $d \in D_B$  bzw.  $m \in M_B$ .

Die mathematischen Referenten einer PRS repräsentieren die im Text verwendeten mathematischen Formeln, Terme sowie freie Variablen. Jede Formel und jeder Term der Formelsprache, sowie jede freie Variable in einer Formel oder und jede Variable in einem Term, wird unter den mathematischen Referenten der Sub-PRS, die den Satz oder das Satzglied dieser Formel oder dieses Terms übersetzt, aufgelistet, sofern die Formel, der Term oder die Variable nicht schon in einer zugänglichen PRS als mathematischer Referent aufgezählt wurde.

Diejenigen mathematischen Referenten, die Variablen sind, nennen wir auch Variablenreferenten. Diejenigen mathematischen Referenten, die Terme sind, nennen wir Termreferenten (so dass die Variablenreferenten auch zu den Termreferenten zählen). Diejenigen mathematischen Referenten, die Formeln sind, nennen wir Formelreferenten.

*math\_id*-Bedingungen ordnen einem Term oder einer Formel einen Diskursreferenten zu. Gewöhnlicherweise steht eine Bedingung der Form *math\_id*( $d, m$ ) in einer PRS  $B$ , für die  $m \in M_B$  gilt. Es kann dann eine Bedingung geben, von der aus  $B$  und damit  $m$  zugänglich ist, *math\_id*( $d, m$ ) aber nicht. Die Zuordnung

---

<sup>†</sup> $\Theta_1$  und  $\Theta_2$  müssen hier als Vorkommnisse (engl. *token*) von PRSen oder PRS-Bedingungen interpretiert werden, nicht als Typen von PRSen oder PRS-Bedingungen: Wenn z.B.  $\Sigma_B = \langle c_1, c_2, c_1 \rangle$ , dann ordnet  $c_2$  das zweite Vorkommnis von  $c_1$ , aber nicht das erste Vorkommnis von  $c_1$ , unter. Es reicht also nicht nur von Typen zu sprechen („ $c_2$  ordnet  $c_1$  direkt unter“), sondern es muss streng genommen von Vorkommnissen gesprochen werden („dieses Vorkommnis von  $c_2$  ordnet das zweite Vorkommnis von  $c_1$  unter“).

von  $d$  zu  $m$  ist aber laut der PRS-Semantik, wie sie im nächsten Abschnitt definiert wird, dann immer noch gegeben. Deswegen macht es Sinn, die Bedingung  $math\_id(d, m)$  noch als „indirekt zugänglich“ zu bezeichnen:

**Definition 3.1.5.** *Indirekte Zugänglichkeit*

Eine Bedingung  $c_1$  ist von einer Bedingung  $c_2$  aus *indirekt zugänglich*, wenn  $c_1$  von  $c_2$  aus zugänglich ist, oder es eine von  $c_2$  aus zugängliche PRS  $B$  gibt, so dass  $c_1 \in \Sigma_B$ .

Häufig ist festgelegt, welche Formelsprache und welche Menge von Prädikatoren in einer PRS Verwendung finden kann:

**Definition 3.1.6.** Sei  $L$  eine Formelsprache und  $\Lambda$  eine Menge von Prädikatoren. Eine PRS  $B$  ist *auf  $(L, \Lambda)$  beschränkt*, wenn die folgenden Bedingungen erfüllt sind:

- Jede *math\_id*-Bedingung  $c$  in  $B$  enthält nur Konstanten-, Funktions- und Relationszeichen, die entweder in  $L$  sind oder von einer von  $c$  aus zugänglichen PRS-Bedingung definiert werden.
- Der Prädikator von jeder Prädikatorbedingung in  $B$  ist in  $\Lambda$  oder wird von einer zugänglichen PRS-Bedingung definiert.
- Keine Definitionsbedingung definiert ein Relationszeichen von  $L$  oder einen Prädikator in  $\Lambda$ .
- Keine Definitionsbedingung  $c$  definiert ein Relationszeichen oder einen Prädikator, der von einer von  $c$  zugänglichen Definitionsbedingung definiert wird.

Die Relationszeichen in  $L$  und die Prädikatoren in  $\Lambda$  sind also die Zeichen, die nicht definiert werden brauchen, sondern als gegeben angenommen werden (wie z.B. „ $\in$ “ in der Mengenlehre).

**Definition 3.1.7.** *PRS-Gebilde*

Als *PRS-Gebilde* bezeichnen wir jede PRS, jede PRS-Bedingung, und jede Folge von PRS-Bedingungen.

Später werden wir mehrere Induktionsbeweise über die Komplexität von PRS-Gebilden durchführen. Dafür wird eine Definition dieser Komplexität benötigt:

**Definition 3.1.8.** *Komplexität von PRS-Gebilden*

Sei  $\Theta$  ein PRS-Gebilde. Dann wird die Komplexität von  $\Theta$ ,  $K(\Theta)$ , wie folgt definiert:

- Ist  $\Theta = holds(d)$ ,  $\Theta = p(d_1, \dots, d_n)$ ,  $\Theta = math\_id(d, m)$  oder  $\Theta = contradiction$  für beliebige Diskursreferenten  $d, d_1, \dots, d_n$ , jeglichen mathematischen Referenten  $m$  und jeglichen Prädikator  $p$ , so ist  $K(\Theta) := 0$ .



- Ist  $\Theta = \neg B$ , so ist  $K(\Theta) := K(B) + 3$ .
- Ist  $\Theta = B \rightarrow C$ ,  $\Theta = B \Rightarrow C$  oder  $\Theta = B := C$ , so ist  $K(\Theta) := K(B) + K(C) + 1$ .
- Ist  $\Theta$  eine Folge  $\langle c_1, \dots, c_n \rangle$  von PRS-Bedingungen, so ist

$$K(\Theta) := \sum_{i=1}^n K(c_i) + n.$$

- Ist  $\Theta$  eine PRS mit PRS-Bedingungs-Folge  $\Sigma_\Theta$ , so ist  $K(\Theta) := K(\Sigma_\Theta) + 1$ .

## 3.2 PRS-Semantik

Bei der Definition der Semantik der Formelsprache im letzten Kapitel haben wir von  $L$ -Strukturen Gebrauch gemacht. Diese geben den Konstanten-, Funktions- und Relationszeichen einer Formelsprache eine Bedeutung. Da die Formelsprache in die Naproche-Sprache eingebettet werden kann, müssen wir auch jetzt mit Strukturen arbeiten, die den Konstanten-, Funktions- und Relationszeichen eine Bedeutung geben. Allerdings müssen diese Strukturen auch den Prädikatoren (Verben, Nomen, Adjektiven) der Naproche-Sprache eine Bedeutung geben:

### Definition 3.2.1. $(L, \Lambda)$ -Struktur

Sei  $L$  eine Formelsprache und  $\Lambda$  eine Menge von Prädikatoren. Eine  $(L, \Lambda)$ -Struktur  $\mathfrak{A}$  mit Universum  $U$  ist ein Paar  $\langle U, \beta \rangle$ , wobei  $U$  eine Menge ist und  $\beta$  eine Abbildung ist, die jedem Konstantenzeichen von  $L$  ein Element in  $U$ , jedem  $n$ -stelligem Funktionszeichen von  $L$  ein Funktion von  $U^n$  nach  $U$ , jedem  $n$ -stelligem Relationszeichen von  $L$  eine Teilmenge von  $U^n$  und jedem  $n$ -stelligem Prädikator in  $\Lambda$  eine Teilmenge von  $U^n$  zuordnet. Für ein Konstantenzeichen  $c$ , ein Funktionszeichen  $f$ , ein Relationszeichen  $R$  und einen Prädikator  $p$  schreiben wir anstatt  $\beta(c)$ ,  $\beta(f)$ ,  $\beta(R)$  bzw.  $\beta(p)$  auch  $\mathfrak{A}(c)$ ,  $\mathfrak{A}(f)$ ,  $\mathfrak{A}(R)$  bzw.  $\mathfrak{A}(p)$ . Wir bezeichnen das Universum von  $\mathfrak{A}$  auch mit  $|\mathfrak{A}|$ .

In einem Naproche-Text können neue Zeichen und Wörter durch Definitionen eingeführt werden. Diese Definitionen werden in der PRS durch Definitionsbedingungen (also Bedingungen der Form  $C := D$ ) dargestellt. Die beabsichtigte Bedeutung einer solchen Definitionsbedingung ist, dass die in Betracht gezogene  $(L, \Lambda)$ -Struktur um ein Relationszeichen oder Prädikator erweitert wird. Dafür müssen wir erstmal definieren, was eine Erweiterung einer  $(L, \Lambda)$ -Struktur ist:

### Definition 3.2.2. Erweiterung von $(L, \Lambda)$

Eine Paar  $(L', \Lambda')$  ist eine *Erweiterung* von  $(L, \Lambda)$  (geschrieben  $(L, \Lambda) \preceq (L', \Lambda')$ ), wenn  $\Lambda \subseteq \Lambda'$  und  $A_L \subseteq A_{L'}$  gilt.  $(L', \Lambda')$  ist eine Erweiterung von  $(L, \Lambda)$  auf eine Menge  $M = \{p_1, \dots, p_n, c_1, \dots, c_k, f_1, \dots, f_l, R_1, \dots, R_m\}$  von Prädikatoren,

Konstanten-, Funktions- und Relationszeichen, wenn  $(L, \Lambda) \preceq (L', \Lambda')$  und  $(\Lambda' \cup A_{L'}) = (M \cup \Lambda \cup A_L)$  gilt.

**Definition 3.2.3.** *Erweiterung einer  $(L, \Lambda)$ -Struktur*

Eine  $(L', \Lambda')$ -Struktur  $\mathfrak{A}' = \langle U', \beta' \rangle$  ist eine *Erweiterung* einer  $(L, \Lambda)$ -Struktur  $\mathfrak{A} = \langle U, \beta \rangle$  (geschrieben  $\mathfrak{A} \preceq \mathfrak{A}'$ ) genau dann, wenn

1.  $(L, \Lambda) \preceq (L', \Lambda')$
2.  $U = U'$
3.  $\beta \subseteq \beta'$ .

Wenn  $(L', \Lambda')$  eine Erweiterung von  $(L, \Lambda)$  auf  $M$  ist, dann nennen wir  $\mathfrak{A}'$  eine Erweiterung von  $\mathfrak{A}$  auf  $M$ .

*holds*-Bedingungen stellen im Naproche-Text gebrauchte Formeln der Formelsprache dar. Um *holds*-Bedingungen zu interpretieren müssen wir uns also auf die Modell-Relation ( $\models$ ) der Formelsprache beziehen. Da wir aber mit einer  $(L, \Lambda)$ -Struktur statt mit einer  $L$ -Struktur arbeiten, müssen wir erst die Modell-Relationen auf  $(L, \Lambda)$ -Strukturen erweitern:

**Definition 3.2.4.** Sei  $\mathfrak{A}$  eine  $(L, \Lambda)$ -Struktur und  $\varphi$  eine  $L$ -Formel. Man beachte, dass es genau eine  $(L, \emptyset)$ -Struktur  $\mathfrak{A}'$  gibt, so dass  $\mathfrak{A}' \preceq \mathfrak{A}$  gilt. Da jede  $(L, \emptyset)$ -Struktur eine  $L$ -Struktur ist, ist  $\mathfrak{A}'$  eine  $L$ -Struktur. Anstelle von  $\mathfrak{A}' \models \varphi$  schreiben wir auch  $\mathfrak{A} \models \varphi$ .

Bei der Definition der Semantik der Formelsprache im letzten Kapitel haben wir von Teilbelegungen Gebrauch gemacht. Diese haben beim Interpretieren einer Formel den Variablen dieser Formel Elemente im Universum der betrachteten Struktur zugeordnet. Bei PRSen haben die Diskursreferenten eine ähnliche Funktion, wie die Variablen in der Formelsprache haben. Die Diskursreferenten beziehen sich aber nicht nur auf Elemente der betrachteten Struktur, sondern auch auf Formeln der Formelsprache. Somit müssen wir beim Interpretieren von PRSen den Diskursreferenten Elemente aus dem Universum der betrachteten Struktur und Formeln der Formelsprache zuordnen. Dies tun wir über partielle Funktionen, die wir wie in der Diskursrepräsentationstheorie *Einbettungen* nennen:

**Definition 3.2.5.** *Einbettung*

Eine *Einbettung* (oder *partielle Funktion*)  $i$  von  $X$  nach  $Y$  ( $i : X \rightsquigarrow Y$ ) ist eine Teilmenge von  $X \times Y$ , so dass für alle  $x \in X$  und  $y_1, y_2 \in Y$ , für die  $\langle x, y_1 \rangle \in i$  und  $\langle x, y_2 \rangle \in i$  gilt,  $y_1 = y_2$  gilt (d.h.  $i$  ist rechteindeutig). Anstelle von  $\langle x, y \rangle \in i$  schreibt man auch  $i(x) = y$ .

**Definition 3.2.6.** *Definitions- und Bildmenge*

Sei  $f$  eine Funktion oder Einbettung. Dann bezeichnen wir mit  $Def(f)$  die Definitionsmenge von  $f$ , also  $\{x : \exists y \langle x, y \rangle \in f\}$ . Mit  $Bild(f)$  bezeichnen wir die Bildmenge von  $f$ , also  $\{y : \exists x \langle x, y \rangle \in f\}$ .

**Definition 3.2.7.** *Erweiterung von Einbettungen*

Eine Einbettung  $j$  ist eine *Erweiterung* der Einbettung  $i$ , falls  $i \subseteq j$  gilt.  $j$  ist eine *Erweiterung* von  $i$  auf  $\{x_1, \dots, x_n\}$ , falls  $j$  eine Erweiterung von  $i$  ist und die Definitionsmenge von  $j$  die Vereinigung von  $\{x_1, \dots, x_n\}$  und der Definitionsmenge von  $i$  ist; hierfür schreiben wir auch  $i[x_1, \dots, x_n]j$ .

Die hier vorgestellte PRS-Semantik ist eine auf PRSen angepasste Version der dynamischen DRS-Semantik, wie sie in [1] vorgestellt wird. In der dynamischen DRS-Semantik steht die Idee der Kontextänderung im Vordergrund: Als Kontexte werden dort Einbettungen von Diskursreferenten in das Universum einer Struktur verwendet. Es wird definiert, wann eine DRS ein Paar von Kontexten  $\langle g, h \rangle$  erfüllt, wobei das intuitive Verständnis dabei ist, dass die DRS den ersten Kontext in den zweiten Kontext ändert.

Bei der darauf basierten PRS-Semantik sind die Kontexte nicht einfach Einbettung von Diskursreferenten in das Universum einer Struktur, sondern Tripel, die aus einer Struktur, einer solchen Einbettung der Diskursreferenten und einer Teilbelegung bestehen. Die Kontextänderung findet nicht nur bei PRSen statt, sondern auch bei PRS-Bedingungen: Dadurch wird es möglich, die Erweiterung der Struktur, die durch eine Definitionsbedingung hervorgerufen wird, als Kontextänderung zu betrachten.

**Definition 3.2.8.** Ein *Kontext* ist ein Tripel  $[\mathfrak{A}, i, \sigma]$ , wobei  $\mathfrak{A}$  eine  $(L, \Lambda)$ -Struktur,  $i \rightsquigarrow F_L \cup |\mathfrak{A}|$  eine Einbettung und  $\sigma$  eine  $|\mathfrak{A}|$ -Teilbelegung ist.

**Definition 3.2.9.** *Dynamische Semantik von PRSen und ihren Bedingungen*

Sei  $L$  eine Formelsprache,  $\Lambda$  eine Menge von Prädikatoren. Seien  $\mathfrak{A}$  und  $\mathfrak{B}$   $(L, \Lambda)$ -Strukturen mit Universum  $U$ ,  $i, j_0: \mathbb{N} \rightsquigarrow F_L \cup U$  Einbettungen und  $\sigma, \tau, \tau_0$   $U$ -Teilbelegung.

Seien  $C$  und  $D$  auf  $(L, \Lambda)$  beschränkte PRSen:

$$C := \begin{array}{|l|l|} \hline d_1, \dots, d_m & v_1, \dots, v_k, m_1, \dots, m_n \\ \hline c_1 & \\ \hline \vdots & \\ \hline c_l & \\ \hline \end{array}$$

$$D := \begin{array}{|l|l|} \hline d'_1, \dots, d'_{m'} & v'_1, \dots, v'_{k'}, m'_1, \dots, m'_{n'} \\ \hline c'_1 & \\ \hline \vdots & \\ \hline c'_{l'} & \\ \hline \end{array}$$

Dabei seien  $v_1, \dots, v_k$  die Variablenreferenten von  $C$ , und  $m_1, \dots, m_n$  die anderen mathematischen Referenten von  $C$ . Entsprechend bei  $D$ .

Analog zu der Schreibweise der dynamischen DRS-Semantik in [1], schreiben wir  $[\mathfrak{A}, i, \sigma], [\mathfrak{B}, j, \tau] \models \Theta$  um auszudrücken, dass die PRS oder PRS-Bedingung

$\Theta$  den Kontext  $[\mathfrak{A}, i, \sigma]$  in den Kontext  $[\mathfrak{B}, j, \tau]$  ändert. Alternativ kann man auch sagen, dass die Kontextänderung von  $[\mathfrak{A}, i, \sigma]$  nach  $[\mathfrak{B}, j, \tau]$   $\Theta$  erfüllt.

Es folgt eine Auflistung aller möglichen Kontextänderungen:

1.  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, j_0, \tau_0] \models C$  genau dann, wenn  $i[d_1, \dots, d_m]j_0$  und  $\sigma[v_1, \dots, v_k]\tau_0$ , und es  $\mathfrak{A}_0, \dots, \mathfrak{A}_l, j_1, \dots, j_l$  und  $\tau_1, \dots, \tau_l$  gibt, so dass  $\mathfrak{A}_0 = \mathfrak{A}$  und  $[\mathfrak{A}_{p-1}, j_{p-1}, \tau_{p-1}], [\mathfrak{A}_p, j_p, \tau_p] \models c_p$  für  $1 \leq p \leq l$ .
2.  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, i, \sigma] \models p(d_1, \dots, d_n)$  genau dann, wenn  $\langle i(d_1), \dots, i(d_n) \rangle \in \mathfrak{A}(p)$ .
3.  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, i, \sigma] \models \mathit{math\_id}(d, m)$  genau dann, wenn  $m$  ein Term ist und  $i(d) = \bar{\sigma}(m)$  gilt oder wenn  $m$  eine Formel von  $L$  ist und  $i(d) = m$  gilt.
4.  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, i, \sigma] \models \mathit{holds}(d)$  genau dann, wenn  $i(d)$  eine Formel ist und  $\mathfrak{A}, \sigma \models i(d)$  gilt.
5.  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, i, \sigma] \models \neg C$  genau dann, wenn es keine Erweiterungen  $j$  von  $i$  und  $\tau$  von  $\sigma$  gibt, so dass  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, j, \tau] \models C$  gilt.
6.  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, i, \sigma] \models C \rightarrow D$  (bzw.  $C \Rightarrow D$ ) genau dann, wenn es für jedes Paar  $[j, \tau]$ , so dass  $[\mathfrak{A}, i, \sigma], [\mathfrak{A}, j, \tau] \models C$  gilt, ein Paar  $[k, v]$  gibt, so dass  $[\mathfrak{A}, j, \tau], [\mathfrak{A}, k, v] \models D$  gilt.
7.  $[\mathfrak{A}, i, \sigma], [\mathfrak{B}, i, \sigma] \models C := D$  genau dann, wenn  $\mathfrak{B}$  eine Erweiterung von  $\mathfrak{A}$  auf das in  $C := D$  definierte Relationszeichen oder den in  $C := D$  definierten Prädikator ist, und für alle  $j_1, \tau_1$  so dass
  - a)  $i[d_1, \dots, d_m]j_1$
  - b)  $\sigma[v_1, \dots, v_k]\tau_1$
  - c)  $[\mathfrak{B}, j_1, \tau_1], [\mathfrak{B}, j_1, \tau_1] \models c'$  für jede  $\mathit{math\_id}$ -Bedingung  $c'$  in  $C$ , $[\mathfrak{B}, i, \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$  genau dann, wenn es  $j_2, \tau_2$  gibt, so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$ .

**Bemerkung.** Da *contradiction* in dieser vollständigen Liste der Kontextänderungen nicht vorkommt, kann keine Kontextänderung eine Bedingung der Form *contradiction* erfüllen.

**Definition 3.2.10.** *Verifikation einer PRS durch eine  $(L, \Lambda)$ -Struktur*

Eine  $(L, \Lambda)$ -Struktur  $\mathfrak{A}$  *verifiziert*  $C$  genau dann, wenn es eine  $|\mathfrak{A}|$ -Teilbelegung  $\sigma$  und eine Einbettung  $i: \mathbb{N} \rightsquigarrow F_L \cup |\mathfrak{A}|$  gibt, so dass  $[\mathfrak{A}, \emptyset, \emptyset], [\mathfrak{A}, i, \sigma] \models C$  gilt.

**Definition 3.2.11.** *Gültigkeit einer PRS*

$C$  ist *gültig* genau dann, wenn jede  $(L, \Lambda)$ -Struktur  $C$  verifiziert.

**Definition 3.2.12.** *Korrektheit eines Naproche-Texts*

Ein Naproche-Text ist *korrekt*, wenn die PRS, in die er übersetzt wird, gültig ist.

**Bemerkung.** Die hier vorgestellte PRS-Semantik sowie die in dieser Arbeit nicht näher behandelte Übersetzung von Naproche-Texten in PRSen wurden mit dem Ziel erstellt, dass diese formale Definition der Korrektheit eines Naproche-Texts mit unserem Korrektheitsempfinden bei informellen Beweisen übereinstimmt.



## Kapitel 4

---

# Das Formelbild einer PRS

---

Es ist möglich, eine PRS in einen Satz der Formelsprache zu übersetzen. Diese Übersetzung, die wir als das *Formelbild* der PRS bezeichnen, gibt den Inhalt der PRS korrekt wieder, was bedeutet, dass das Formelbild von einer PRS  $C$  genau in jenen Strukturen wahr ist, die  $C$  verifizieren.

Es hat nicht jede PRS ein Formelbild, sondern es müssen einige zusätzliche Bedingungen erfüllt sein:

**Definition 4.0.1.** Ein *wohlgeformtes* PRS-Gebilde ist ein PRS-Gebilde  $\Theta$ , für das gilt:

1. Jeder Diskursreferent, der in einer *math\_id*-Bedingung  $c_1$  oder Prädikatorbedingung  $c_2$  in einer Sub-PRS von  $\Theta$  vorkommt, ist von  $c_1$  bzw.  $c_2$  aus zugänglich.
2. Für jede Sub-PRS  $B$  von  $\Theta$  und jede Bedingung  $c \in \Sigma_B$  der Form *math\_id*( $d, m$ ) gilt  $m \in M_B$ .
3. Wenn eine Bedingung *math\_id*( $d, m$ ) von einer Bedingung *math\_id*( $d', m'$ ) aus indirekt zugänglich ist, dann ist  $d \neq d'$ .
4. Für jede Bedingung  $c$  der Form *math\_id*( $d, m$ ) und jede Variable  $x \in \text{frei}(m)$  gibt es eine von  $c$  aus indirekt zugänglich Bedingung der Form *math\_id*( $d', x$ ).
5. Für jede Bedingung der Form *holds*( $d$ ) muss es eine indirekt zugängliche Bedingung der Form *math\_id*( $d, \varphi$ ) für eine Formel  $\varphi$  geben.
6. Für jede Bedingung  $c$  der Form  $p(d_1, \dots, d_n)$  gibt es keine von  $c$  aus indirekt zugängliche Bedingung der Form *math\_id*( $d_i, \varphi$ ) für  $1 \leq i \leq n$  und für eine Formel  $\varphi$ .

7. Für jede Sub-PRS  $B$  von  $\Theta$ , jedes  $d \in D_B$  und jedes  $m \in M_B$ , sind  $d$  und  $m$  von  $B$  aus nicht zugänglich.

## 4.1 Lambda-Kalkül

Bei der Definition des Formelbildes machen wir vom Lambda-Kalkül Gebrauch. Das Lambda-Kalkül soll hier nicht als Sprache für Logik höherer Ordnung verstanden werden, sondern nur als eine praktische Notation für durchzuführende Substitutionen. Genauer gesagt sollen Lambda-Ausdrücke als Funktionen von Termen nach Formeln einer Formelsprache angesehen werden. Für die Zwecke dieser Arbeit ist die folgende Definition hinreichend:

**Definition 4.1.1.** *Lambda-Ausdrücke*

Sei  $L$  eine Formelsprache,  $\varphi$  eine  $L$ -Formel und  $x$  ein Variablenzeichen von  $L$ . Dann ist  $\lambda x.\varphi$  ein  $L$ -Lambda-Ausdruck. Ein  $L$ -Lambda-Ausdruck ist eine Funktion von  $L$ -Termen nach  $L$ -Formeln, die wie folgt definiert wird:

$$\lambda x.\varphi(t) := \varphi \frac{t}{x}$$

Alle Vorkommnisse von  $x$  in  $\lambda x.\varphi$  gelten als gebunden. Wir können also  $\text{frei}(\lambda x.\varphi)$  als  $\text{frei}(\varphi) \setminus \{x\}$  definieren.

## 4.2 Definition des Formelbildes

Für den Rest dieser Arbeit sei  $\Lambda$  eine Menge von Prädikatoren. Von nun an werden alle PRSen als auf  $(L, \Lambda)$  beschränkt angenommen.

Das Formelbild einer PRS wird in einer erweiterten Version der Formelsprache  $L$  ausgedrückt. Und zwar wird für jeden Prädikator in  $\Lambda$  ein neues Relationszeichen in  $L$  eingeführt. Anders ausgedrückt führen wir für jedes Verb, Nomen und Adjektiv der Naproche-Sprache ein formelsprachliches Relationszeichen ein. Der Einfachheit halber gehen wir davon aus, dass  $\Lambda$  und die Menge der Relationszeichen von  $L$  disjunkt sind, so dass wir die Prädikatoren ohne Umbenennung als Relationszeichen der Formelsprache benutzen können. Zusätzlich erweitern wir  $L$  auch noch um eine Folge  $\delta_0, \delta_1, \delta_2, \dots$  von neuen Variablenzeichen, die bei der Übersetzung einer PRS für die Übersetzung der Diskursreferenten, die nicht durch *math.id*-Bedingungen an  $L$ -Terme gebunden werden, genutzt werden. Diese Erweiterung von  $L$  nennen wir  $L'$ .

**Definition 4.2.1.** Wir definieren die Funktion  $\delta$  von der Menge der Diskursreferenten und Variablenzeichen von  $L$  nach der Menge der Variablenzeichen von  $L'$  durch

$$\delta(x) := \begin{cases} \delta_x & \text{wenn } x \text{ ein Diskursreferent ist.} \\ x & \text{wenn } x \text{ ein Variablenzeichen von } L \text{ ist.} \end{cases}$$

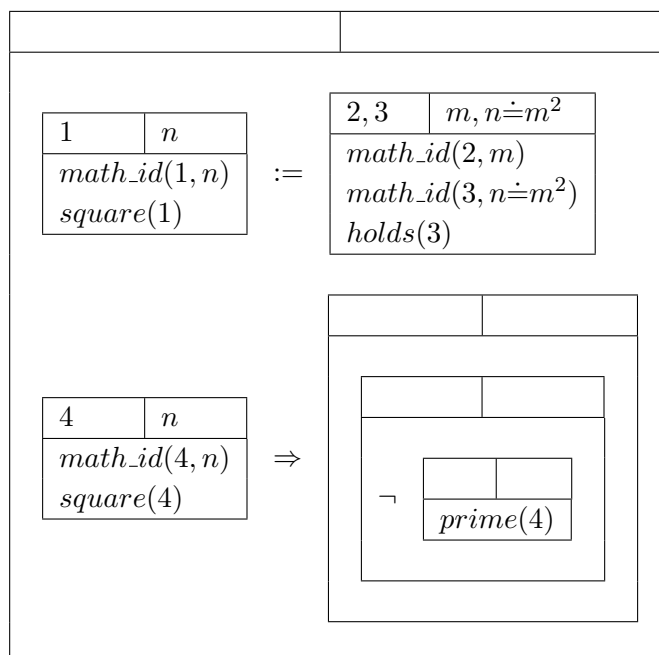


Auch in KÜHLWEIN [9] wurde schon ein Formelbild definiert, wobei dieses allerdings eine Folge von prädikatenlogischen Formeln war, und nicht ein einziger prädikatenlogischer Satz. Da zu dem Zeitpunkt jener Diplomarbeit allerdings noch keine Definition der PRS-Semantik gegeben war, konnte dieses Formelbild auch nicht den Anspruch einer korrekten Übersetzung der PRS haben; denn die Korrektheit einer Übersetzung lässt sich nur charakterisieren, wenn für das zu Übersetzende eine festgelegte Bedeutung existiert. Mit der Festlegung der PRS-Semantik wurde allerdings klar, dass die Definitionsbedingungen und die von ihnen definierten Relationszeichen im Formelbild anders behandelt werden müssen als in KÜHLWEINS Formelbild, wenn das Ziel einer korrekten Übersetzung erreicht werden soll. Denn in KÜHLWEINS Formelbild wird eine Definitionsbedingung durch einen Bikonditional zwischen dem Definiendum und dem Definiens übersetzt, welcher jedoch in einer Struktur, in der dem Definiendum eine andere Menge zugeordnet ist als dem Definiens, nicht erfüllt wird. Daher definieren wir das Formelbild jetzt auf eine solche Weise, dass eine Definitionsbedingung überhaupt nicht übersetzt wird, sondern zu einer Substitution des Definiendum durch das Definiens in der Übersetzung der ihr folgenden Bedingungen führt.

Zum Beispiel wird der folgende Text:

Define  $n$  to be square if and only if there is an  $m$  such that  $n \doteq m^2$ .  
Let  $n$  be square. Then  $n$  is not prime.

in die folgende PRS übersetzt:



In dem Formelbild dieser PRS erscheint die Definitionsbedingung nicht mehr. Stattdessen führt die Definitionsbedingung dazu, dass „square“ im Formelbild nicht mehr erscheint, sondern durch sein Definiens ersetzt wird. Wie später noch detailliert erklärt wird, ist das Formelbild dieser PRS

$$\forall n(\exists m n \doteq m^2 \rightarrow \neg \text{prime}(n)).$$

Hierbei wurde also nur die zweite Bedingung wirklich übersetzt, wobei die erste Bedingung nur zu einer Ersetzung von  $\text{square}(n)$  durch  $\exists m n \doteq m^2$  in der zweiten Bedingung geführt hat.

Um das Formelbild einer PRS zu definieren, müssen wir rekursiv das Formelbild von PRS-Bedingungen, Folgen von PRS-Bedingungen und PRSen definieren. Um die oben erwähnte Substitution der Definienda zu verwirklichen, müssen aber in jedem Schritt der Rekursion Informationen zu den anzuwendenden Substitutionen für den nächsten Schritt gespeichert werden. Um das zu erreichen, erhält die Formelbild-Funktion, mittels derer die Formelbilder von PRS-Bedingungen und Folgen von PRS-Bedingungen definiert werden, ein zusätzliches Argument, welches eine Funktion enthält, die die durchzuführenden Substitutionen festlegt.

Diese Funktion, die wir durch den Buchstaben  $\Delta$  kennzeichnen, ordnet jedem zu ersetzenden Relationszeichen ein Lambda-Ausdruck zu. Wenn z.B.  $\text{square}(n)$  definiert wurde als  $\exists m n \doteq m^2$ , so wird  $\Delta$  dem Relationszeichen  $\text{square}$  den Lambda-Ausdruck  $\lambda n. \exists m n \doteq m^2$  zuordnen. Durch die Verwendung des Lambda-Kalküls wird hier gewährleistet, dass die Argumente der definierten Relation nach der Substitution an korrekter Stelle im Definiens auftreten.

**Definition 4.2.2.** *Substitution von Relationszeichen*

Eine solche Funktion  $\Delta$  definiert eine Funktion  $\bar{\Delta}$  von  $L'$ -Formeln nach  $L'$ -Formeln, die die gewünschte Substitution auf  $L'$ -Formeln durchführt:

1.  $\bar{\Delta}(R(t_1, \dots, t_n)) := \Delta(R)(t_1) \dots (t_n)$  für jedes Relationszeichen\*  $R \in \text{Def}(\Delta)$  und jegliche  $L'$ -Terme  $t_1, \dots, t_n$ .
2.  $\bar{\Delta}(R(t_1, \dots, t_n)) := R(t_1, \dots, t_n)$  für jedes Relationszeichen  $R \notin \text{Def}(\Delta)$  und jegliche  $L'$ -Terme  $t_1, \dots, t_n$ .
3. Für  $L'$ -Formeln  $\varphi$  und  $\psi$  gilt
  - $\bar{\Delta}(\neg\varphi) := \neg\bar{\Delta}(\varphi)$
  - $\bar{\Delta}((\varphi \wedge \psi)) := (\bar{\Delta}(\varphi) \wedge \bar{\Delta}(\psi))$
  - $\bar{\Delta}((\varphi \vee \psi)) := (\bar{\Delta}(\varphi) \vee \bar{\Delta}(\psi))$
  - $\bar{\Delta}((\varphi \rightarrow \psi)) := (\bar{\Delta}(\varphi) \rightarrow \bar{\Delta}(\psi))$

---

\*Hier sind Relationszeichen von  $L'$  gemeint. Dies sind also sowohl die Relationszeichen von  $L$  als auch die Prädikatoren in  $\Lambda$

- $\bar{\Delta}((\varphi \leftrightarrow \psi)) := (\bar{\Delta}(\varphi) \leftrightarrow \bar{\Delta}(\psi))$

Zusätzlich zu den Informationen zu den durchzuführenden Substitutionen von Relationszeichen müssen auch Informationen zu den durch *math.id*-Bedingungen festgelegten Interpretationen der Diskursreferenten gespeichert werden. Dies machen wir durch partielle Funktionen von Diskursreferenten nach mathematischen Referenten, die wir durch die Buchstaben  $k$  und  $l$  kennzeichnen, und die als zusätzliche Argumente in der Formelbildfunktion verwendet werden.

Bei der Übersetzung einer PRS oder Sub-PRS  $B$  muss außerdem über alle in  $B$  eingeführten Variablen quantifiziert werden. Dafür brauchen wir eine Definition der „eingeführten“ Variablen:

**Definition 4.2.3.** Sei  $B$  eine PRS mit Diskursreferenten  $d_1, \dots, d_n$  und Variablenreferenten  $v_1, \dots, v_n$ . Sei  $l$  eine partielle Funktion von Diskursreferenten nach  $L$ -Formeln und  $L$ -Termen.

- $Y_{B,l} := \{\delta_d : d \in \{d_1, \dots, d_n\} \setminus Def(l)\}$
- $Z_B := \{v_1, \dots, v_n\}$
- $X_{B,l} := Y_{B,l} \cup Z_B$

**Definition 4.2.4.** Wenn  $X$  eine Menge  $\{x_1, \dots, x_n\}$  von Variablen ist, so schreiben wir  $\exists X$  als Kurzform für  $\exists x_1 \dots \exists x_n$ ; analog für  $\forall$ .  $\exists \emptyset$  und  $\forall \emptyset$  stehen dann logischerweise für leere Zeichenfolgen. (Die Reihenfolge der  $x_1 \dots x_n$  spielt dabei semantisch keine Rolle. Zur Wohldefiniertheit dieser Definition müssen wir aber streng genommen eine Reihenfolge eindeutig festlegen. Dies kann allerdings problemlos gemacht werden, da die Menge aller Variablenzeichen von  $L'$  abzählbar ist.)

**Definition 4.2.5. Formelbild**

Durch simultane Rekursion werden drei partielle Formelbildfunktionen definiert: Eine für PRS-Bedingungen (FB), eine für Folgen von PRS-Bedingungen (FF) und eine für PRSen (FP).

Seien  $k, l, l'$  und  $\Delta$  partielle Funktionen. Bei den nun folgenden Definitionsgleichungen ist die linke Seite genau für jene Werte von  $k, l, l'$  und  $\Delta$  definiert, für die die rechte Seite definiert ist.

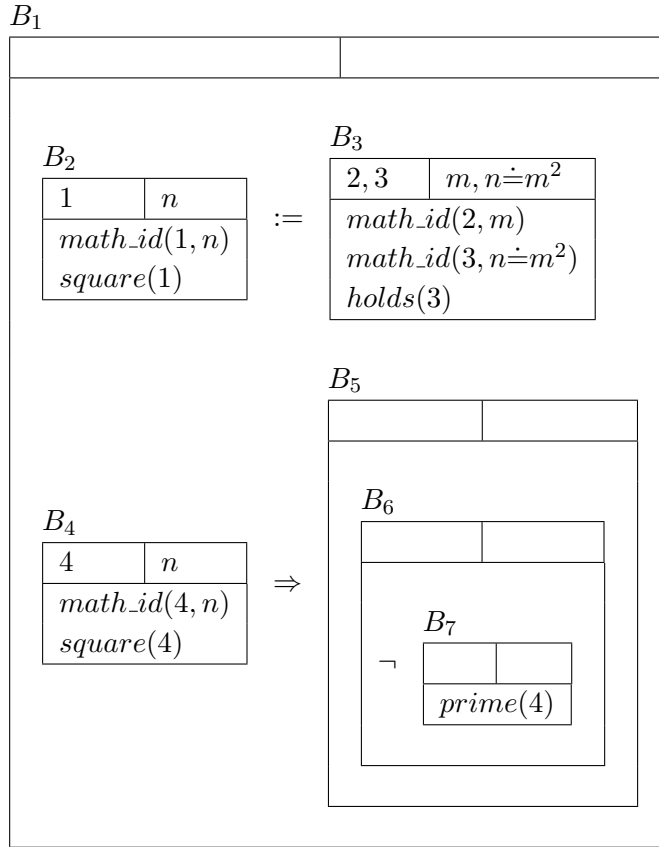
1.  $FB(holds(d), \Delta, k) := \bar{\Delta}(k(d))$ , wenn  $k(d)$  eine Formel ist.
2.  $FB(p(d_1, \dots, d_n), \Delta, k) := \bar{\Delta}(p(\hat{k}(d_1), \dots, \hat{k}(d_n)))$ , wenn  $k(d_i)$  für  $1 \leq i \leq n$  keine Formel ist, wobei  $\hat{k}(d) := \begin{cases} k(d), & \text{wenn } d \in Def(k) \\ \delta_d & \text{ansonsten} \end{cases}$
3.  $FB(contradiction, \Delta, k) := \perp$

4.  $FB(\neg B, \Delta, k) := \neg FF(\langle B \rangle, \Delta, k, k)$
5.  $FB(B \Rightarrow C, \Delta, k) := \forall X_{B,l}(FF(\Sigma_B, \Delta, k, l) \rightarrow FF(\langle C \rangle, \Delta, l, l))$
6.  $FF(\langle \mathit{math\_id}(d, m) \rangle, \Delta, k, k \cup \{d, m\}) := \top$
7.  $FF(\langle \mathit{math\_id}(d, m) \rangle + \Sigma, \Delta, k, l) := FF(\Sigma, \Delta, k \cup \{d, m\}, l)$  für  $\Sigma \neq \langle \rangle$
8.  $FF(\langle C := D \rangle, \Delta, k, k) := \top$
9.  $FF(\langle C := D \rangle + \Sigma, \Delta, k, l) := FF(\Sigma, \Delta', k, l)$ , wenn  $\Sigma \neq \langle \rangle$  und  $FF(\Sigma_C, \Delta, k, k')$  definiert ist, und entweder
  - $C := D$   $R(x_1, \dots, x_n)$  definiert, in welchem Falle wir  $\Delta' := \Delta \cup \{\langle R, \lambda x_n \dots \lambda x_1. FF(\langle D \rangle, \Delta, k', k') \rangle\}$  setzen, oder
  - $C := D$   $p(d_1, \dots, d_n)$  mittels  $\alpha_1, \dots, \alpha_n$  definiert, in welchem Falle wir  $\Delta' := \Delta \cup \{\langle p, \lambda \delta(\alpha_n) \dots \lambda \delta(\alpha_1). FF(\langle D \rangle, \Delta, k', k') \rangle\}$  setzen.
10.  $FF(\langle B \rangle, \Delta, k, k) := \exists X_{B,l} FF(\Sigma_B, \Delta, k, l)$
11.  $FF(\langle B \rangle + \Sigma, \Delta, k, l) := \exists X_{B,l'}(FF(\Sigma_B, \Delta, l, l') \wedge FF(\Sigma, \Delta, k \cup (l' \setminus l), l'))$ , wenn  $\Sigma \neq \langle \rangle$  und  $k \cap (l' \setminus l) = \emptyset$ .
12.  $FF(\langle c \rangle, \Delta, k, k) := FB(c, \Delta, k)$  für eine PRS-Bedingung  $c$ , die nicht PRS, Definitionsbedingung oder  $\mathit{math\_id}$ -Bedingung ist
13.  $FF(\langle c \rangle + \Sigma, \Delta, k, l) := (FB(c, \Delta, l) \wedge FF(\Sigma, \Delta, k, l))$  für  $\Sigma \neq \langle \rangle$  und für eine PRS-Bedingung  $c$ , die nicht PRS, Definitionsbedingung oder  $\mathit{math\_id}$ -Bedingung ist
14.  $FF(\langle \rangle, \Delta, k, k) := \top$
15.  $FP(B) := FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)$

### 4.3 Beispiel einer Formelbild-Konstruktion

Da es nicht leicht ersichtlich ist, was genau der Zweck der zusätzlichen Argumente und der einzelnen Punkte in der obigen Formelbild-Definition ist, soll dies anhand von zwei Beispielen erläutert werden, noch bevor wir im nächsten Abschnitt zeigen, dass das Formelbild wohldefiniert ist und dass  $FP(B)$  für jede wohlgeformte PRS  $B$  definiert ist.

Die oben schon erwähnte PRS wird hier nochmal mit Identifikatoren dargestellt, damit wir uns in den Erläuterungen leichter auf bestimmte Sub-PRSen beziehen können:



Wir zeigen nun, dass  $FP(B_1) = \forall n(\exists m n \doteq m^2 \rightarrow \neg prime(n))$ :

$$\begin{aligned}
 FP(B_1) &= FF(\langle B_1 \rangle, \emptyset, \emptyset, \emptyset) && \text{(Punkt 15 von Definition 4.2.5)} \\
 &= \exists X_{B_1, l} FF(\Sigma_{B_1}, \emptyset, \emptyset, l) && \text{(für ein } l, \text{ für das } FF(\Sigma_{B_1}, \emptyset, \emptyset, l) \text{ definiert} \\
 &&& \text{ist; Punkt 10 von Definition 4.2.5)}
 \end{aligned}$$

Man beachte, dass  $\Sigma_{B_1}$  die Folge der Bedingungen von  $B_1$  ist, also  $\langle B_2 := B_3, B_4 \Rightarrow B_5 \rangle$ . Nun müssen wir bestimmen, für welches  $l$  das Formelbild  $FF(\Sigma_B, \emptyset, \emptyset, l)$  definiert ist, und welchen Wert  $FF(\Sigma_B, \emptyset, \emptyset, l)$  annimmt:

$$FF(\Sigma_{B_1}, \emptyset, \emptyset, l) = FF(\langle B_4 \Rightarrow B_5 \rangle, \Delta, \emptyset, l) \quad \text{(Punkt 9)} \quad (4.1)$$

Hier ist  $\Delta = \{\langle square, \lambda n. FF(\langle B_3 \rangle, \emptyset, k', k') \rangle\}$ , und  $k'$  ist so gewählt, dass  $FF(\Sigma_{B_2}, \emptyset, \emptyset, k')$  definiert ist. Aufgrund von Punkt 7 ist  $FF(\Sigma_{B_2}, \emptyset, \emptyset, k')$  definiert, wenn  $FF(\langle square(1) \rangle, \emptyset, \{ \langle 1, n \rangle \}, k')$  definiert ist, wofür wegen Punkt 12  $k' = \{ \langle 1, n \rangle \}$  gelten muss. Dann gilt aufgrund von Punkt 10

$$FF(\langle B_3 \rangle, \emptyset, k', k') = \exists X_{B_3, l'} FF(\Sigma_{B_3}, \emptyset, k', l') \quad (4.2)$$

für ein  $l'$ , für das die rechte Seite definiert ist. Weiterhin gilt

$$\begin{aligned}
FF(\Sigma_{B_3}, \emptyset, k', l') &= FF(\langle \mathit{math\_id}(2, m), \mathit{math\_id}(3, n \doteq m^2), \mathit{holds}(3) \rangle, \\
&\quad \emptyset, \{\langle 1, n \rangle\}, l') \\
&= FF(\langle \mathit{math\_id}(3, n \doteq m^2), \mathit{holds}(3) \rangle, \emptyset, \{\langle 1, n \rangle, \langle 2, m \rangle\}, l') \\
&\quad \text{(Punkt 7)} \\
&= FF(\langle \mathit{holds}(3), \emptyset, \{\langle 1, n \rangle, \langle 2, m \rangle, \langle 3, n \doteq m^2 \rangle\} \rangle, l') \\
&\quad \text{(nochmals Punkt 7)} \\
&= FB(\mathit{holds}(3), \emptyset, \{\langle 1, n \rangle, \langle 2, m \rangle, \langle 3, n \doteq m^2 \rangle\}) \\
&\quad \text{(Punkt 12; somit gilt } l' = \{\langle 1, n \rangle, \langle 2, m \rangle, \langle 3, n \doteq m^2 \rangle\}) \\
&= \bar{\emptyset}(n \doteq m^2) \quad \text{(Punkt 1)} \\
&= n \doteq m^2
\end{aligned}$$

Hieran sieht man sehr gut die Funktion der beiden letzten Argumente der vierstelligen Funktion  $FF$ : Während man eine Folge von Bedingungen durcharbeitet, wird zu dem vorletzten Argument bei jeder  $\mathit{math\_id}$ -Bedingung die Information dieser  $\mathit{math\_id}$ -Bedingung hinzugefügt. Das letzte Argument bleibt dabei konstant, und enthält die gesamte Information, die am Ende der Bedingungsfolge im vorletzten Argument ist, also die gesamte Information aller  $\mathit{math\_id}$ -Bedingungen in der Bedingungsfolge. Die Information aus diesem letzten Argument wird an die Funktion  $FB$  übergeben, und dann bei der Übersetzung von  $\mathit{holds}$ -Bedingungen und Prädikatorbedingungen zur Auflösung der Diskursreferenten verwendet.

Da wir jetzt den Wert von  $l'$  kennen, können wir auch  $X_{B_3, l'}$  berechnen:  $X_{B_3, l'} = Y_{B_3, l'} \cup Z_{B_3} = \{\delta_d : d \in \{2, 3\} \setminus \mathit{Def}(l')\} \cup \{m\} = \{m\}$ , da  $\mathit{Def}(l') = \{1, 2, 3\}$ . Somit folgt aus 4.2, dass  $FF(\langle B_3 \rangle, \emptyset, k', k')$  die Formel  $\exists m n \doteq m^2$  ist.

Dadurch folgt, dass  $\Delta = \{\langle \mathit{square}, \lambda n. \exists m n \doteq m^2 \rangle\}$ .  $\Delta$  enthält jetzt also die Information, dass wir für jeden Term  $t$   $\mathit{square}(t)$  durch  $\exists m t \doteq m^2$  ersetzen können.<sup>†</sup>

Jetzt können wir die Gleichung 4.1 weiterführen:

$$\begin{aligned}
FF(\Sigma_{B_1}, \emptyset, \emptyset, l) &= FF(\langle B_4 \Rightarrow B_5 \rangle, \Delta, \emptyset, l) \\
&= FB(B_4 \Rightarrow B_5, \Delta, \emptyset) \\
&\quad \text{(Punkt 12; somit gilt } l = \emptyset) \\
&= \forall X_{B_4, l_0} (FF(\Sigma_{B_4}, \Delta, \emptyset, l_0) \rightarrow FF(\langle B_5 \rangle, \Delta, l_0, l_0)) \quad (4.3) \\
&\quad \text{für ein } l_0, \text{ für das dieser Ausdruck definiert ist} \\
&\quad \text{(Punkt 5)}
\end{aligned}$$

Als nächstes müssen wir also  $FF(\Sigma_{B_4}, \Delta, \emptyset, l_0)$  berechnen und dabei  $l_0$  bestim-

<sup>†</sup>sofern  $t$  nicht  $m$  enthält. Wenn  $t$   $m$  enthält, dann ist  $\lambda n. \exists m n \doteq m^2 (t)$  per definitionem gleich  $(\exists m n \doteq m^2) \frac{t}{n}$ , also aufgrund der Definition der Substitution bei quantifizierten Formeln (EBBINGHAUS [3] et al., S. 55) gleich  $\exists u t \doteq u^2$  für eine Variable  $u$ , die nicht in  $t$  vorkommt.

men:

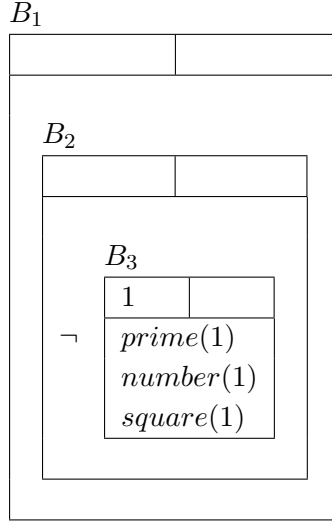
$$\begin{aligned}
FF(\Sigma_{B_4}, \Delta, \emptyset, l_0) &= FF(\langle \text{square}(4) \rangle, \Delta, \{\langle 4, n \rangle\}, l_0) \quad (\text{Punkt 7}) \\
&= FB(\text{square}(4), \Delta, \{\langle 4, n \rangle\}) \quad (\text{Punkt 12; ergo } l_0 = \{\langle 4, n \rangle\}) \\
&= \bar{\Delta}(\text{square}(n)) \quad (\text{Punkt 2}) \\
&= \Delta(\text{square})(n) \quad (\text{aufgrund von Definition 4.2.2, Punkt 1}) \\
&= \lambda n. \exists m \, n \doteq m^2 \, (n) \\
&= \exists m \, n \doteq m^2
\end{aligned}$$

Außerdem müssen wir  $FF(\langle B_5 \rangle, \Delta, l_0, l_0)$  bestimmen:

$$\begin{aligned}
FF(\langle B_5 \rangle, \Delta, l_0, l_0) &= \exists X_{B_5, l_1} FF(\langle B_6 \rangle, \Delta, l_0, l_1) \\
&\quad \text{für ein } l_1, \text{ für das } FF(\langle B_6 \rangle, \Delta, l_0, l_1) \text{ definiert ist} \\
&\quad (\text{Punkt 10}) \\
&= FF(\langle B_6 \rangle, \Delta, l_0, l_1) \\
&\quad (\text{da } X_{B_5, l_1} = \emptyset \text{ unabhängig von } l_1 \text{ gilt}) \\
&= \exists X_{B_6, l_2} FF(\langle \neg B_7 \rangle, \Delta, l_1, l_2) \\
&\quad \text{für ein } l_2, \text{ für das } FF(\langle \neg B_7 \rangle, \Delta, l_1, l_2) \text{ definiert ist} \\
&\quad (\text{Punkt 10; ergo } l_1 = l_0) \\
&= FF(\langle \neg B_7 \rangle, \Delta, l_1, l_2) \\
&\quad (\text{da } X_{B_6, l_2} = \emptyset \text{ unabhängig von } l_2 \text{ gilt}) \\
&= FB(\neg B_7, \Delta, l_0) \quad (\text{Punkt 12; ergo } l_2 = l_0) \\
&= \neg FF(\langle B_7 \rangle, \Delta, l_0, l_0) \quad (\text{Punkt 4}) \\
&= \neg \exists X_{B_7, l_3} FF(\langle \text{prime}(4) \rangle, \Delta, l_0, l_3) \\
&\quad \text{für ein } l_3, \text{ für das } FF(\langle \text{prime}(4) \rangle, \Delta, l_0, l_3) \text{ definiert ist} \\
&\quad (\text{Punkt 10}) \\
&= \neg FF(\langle \text{prime}(4) \rangle, \Delta, l_0, l_3) \\
&\quad (\text{da } X_{B_7, l_3} = \emptyset \text{ unabhängig von } l_3 \text{ gilt}) \\
&= \neg FB(\text{prime}(4), \Delta, l_0) \quad (\text{Punkt 12; ergo } l_3 = l_0) \\
&= \neg \bar{\Delta}(\text{prime}(n)) \quad (\text{Punkt 1; denn } l_0(4) = n) \\
&= \neg \text{prime}(n) \quad (\text{da } \text{prime} \notin \text{Def}(\Delta))
\end{aligned}$$

Weiterhin gilt  $X_{B_4, l_0} = Y_{B_4, l_0} \cup Z_{B_4} = \{\delta_d : d \in \{4\} \setminus \text{Def}(l_0)\} \cup \{n\} = \{n\}$ , denn  $\text{Def}(l_0) = \{4\}$ . Somit können wir nun aus Gleichung 4.3 folgern, dass  $FF(\Sigma_{B_1}, \emptyset, \emptyset, l) = \forall n (\exists m \, n \doteq m^2 \rightarrow \neg \text{prime}(n))$ . Da  $X_{B_1, l} = \emptyset$ , folgt nun, dass  $FP(B_1) = \forall n (\exists m \, n \doteq m^2 \rightarrow \neg \text{prime}(n))$ , was die Berechnung des Formelbildes von  $B_1$  abschließt.

Zusätzlich zu diesem detailliert erläuterten Beispiel sei noch ein Beispiel aufgeführt, in dem eine Variable der Form  $\delta_i$  im Formelbild auftaucht: Der Naproche-Text „*No prime number is square.*“ wird in die folgende PRS übersetzt:



Es folgt die Berechnung des Formelbildes:

$$\begin{aligned}
FB(B_1) &= FF(\langle B_1 \rangle, \emptyset, \emptyset, \emptyset) \\
&= FF(\langle B_2 \rangle, \emptyset, \emptyset, l) \quad (\text{da } X_{B_1, l} = \emptyset) \\
&= FF(\langle \neg B_3 \rangle, \emptyset, l_0) \quad (\text{da } X_{B_2, l_0} = \emptyset) \\
&= FB(\neg B_3, \emptyset, \emptyset) \\
&= \neg FF(B_3, \emptyset, \emptyset) \\
&= \neg \exists X_{B_3, l_1} FF(\Sigma_{B_3}, \emptyset, \emptyset, l_1) \\
&= \neg \exists \delta_1 FF(\Sigma_{B_3}, \emptyset, \emptyset, l_1) \\
&\quad (\text{denn } X_{B_3, l_1} = \{\delta_1\}, \text{ wie unten gezeigt wird}) \\
&= \neg \exists \delta_1 (FB(\text{prime}(1), \emptyset, l_1) \wedge FF(\langle \text{number}(1), \text{square}(1) \rangle, \emptyset, \emptyset, l_1)) \\
&\quad (\text{Punkt 13 von Definition 4.2.5}) \\
&= \neg \exists \delta_1 (\text{prime}(\delta_1) \wedge FF(\langle \text{number}(1), \text{square}(1) \rangle, \emptyset, \emptyset, l_1)) \\
&\quad (\text{Punkt 2; denn } \widehat{l}_1(1) = \delta_1, \text{ wie unten gezeigt wird}) \\
&= \neg \exists \delta_1 (\text{prime}(\delta_1) \wedge (FB(\text{number}(1), \emptyset, l_1) \wedge FF(\langle \text{square}(1) \rangle, \emptyset, \emptyset, l_1))) \\
&\quad (\text{Punkt 13}) \\
&= \neg \exists \delta_1 (\text{prime}(\delta_1) \wedge (\text{number}(\delta_1) \wedge \text{square}(\delta_1))) \\
&\quad (\text{Punkt 12, aus dem auch } l_1 = \emptyset \text{ folgt, und zweimal Punkt 2})
\end{aligned}$$

Die folgenden Ergebnisse waren für diese Berechnung notwendig:



- $X_{B_3, l_1} = Y_{B_3, l_1} \cup Z_{B_3} = (\{\delta_d : d \in \{1\}\} \setminus \text{Def}(l_1)) \cup \emptyset = \{\delta_1\} \setminus \emptyset = \{\delta_1\}$ .
- $\widehat{l}_1(1) = \delta_1$ , da  $l_1(1)$  nicht definiert ist.

Wie schon am Anfang des Kapitels erläutert wurde, ist der Zweck der Variablen der Form  $\delta_i$ , als Übersetzung von Diskursreferenten zu dienen, die nicht durch *math\_id*-Bedingungen an  $L$ -Terme gebunden werden. Das obige Beispiel hat gezeigt, wie die Definition von  $X_{B,l}$  und Punkt 2 der Formelbilddefinition zu dem Auftreten solcher Variablen im Formelbild führen.

## 4.4 Wohldefiniertheit des Formelbildes

Man beachte, dass die drei Formelbildfunktionen partielle Funktionen sind: Sie sind also nur für einige der möglichen Werte ihrer Argumente definiert.

Für die Wohldefiniertheit muss gezeigt werden, dass keine der drei Formelbildfunktionen für festgelegte Argumente mehr als einen Wert annehmen kann. Diese Gefahr besteht deswegen, weil in den Punkten 5, 10 und 11 der Definition auf der rechten Seite der Definitionsgleichung Funktionen ( $l$  oder  $l'$ ) vorkommen, die auf der linken Seite nicht vorkommen. Somit besteht die Gefahr, dass die rechte Seite für mehr als einen Wert dieser Funktionen definiert ist, und dadurch verschiedenen Werte annehmen kann. Das folgende Theorem wendet diese Gefahr ab:

**Theorem 4.4.1.** *Sei  $\Theta$  ein PRS-Gebilde.*

1. *Durch die Gleichungen in der Formelbilddefinition (4.2.5) ist für jedes  $\Delta$ , jedes  $k$  und jedes  $l$  den Formeln  $FB(\Theta, \Delta, k)$ ,  $FF(\Theta, \Delta, k, l)$  und  $FP(\Theta)$  höchstens ein Wert zugeordnet.*
2. *Wenn  $FF(\Theta, \Delta, k, l)$  ein Wert zugeordnet ist, dann ist  $k \subseteq l$ .*
3. *Wenn sowohl  $FF(\Theta, \Delta, k_1, l_1)$  als auch  $FF(\Theta, \Delta, k_2, l_2)$  ein Wert zugeordnet ist, dann ist  $l_1 \setminus k_1 = l_2 \setminus k_2$ .*
4. *Wenn sowohl  $FF(\Theta, \Delta, k, l_1)$  als auch  $FF(\Theta, \Delta, k, l_2)$  ein Wert zugeordnet ist, dann ist  $l_1 = l_2$ .*

### Beweis

Der Beweis verfolgt über eine Induktion über die Komplexität von  $\Theta$ . Der vierte Punkt des Theorems folgt direkt aus dem zweiten und dritten Punkt. Er braucht also nicht gesondert bewiesen zu werden, wird hier aber aufgeführt, da er im Induktionsschritt des Beweises mehrfach verwendet wird.

### Induktionsanfang

Sei  $\Theta$  ein PRS-Gebilde der Komplexität 0. Es können zwei Fälle unterschieden werden:

Fall 1:  $\Theta$  hat die Form  $holds(d)$ ,  $p(d_1, \dots, d_n)$ ,  $math\_id(d, m)$  oder  $contradiction$ .

1. Für  $math\_id(d, m)$  ist kein Formelbild definiert, wohingegen für die anderen drei möglichen Formen von  $\Theta$  der Formel  $FB(\Theta, \Delta, k)$  durch Punkt 1, 2 bzw. 3 von Definition 4.2.5 ein eindeutiger Werte zugeordnet ist, und den Formeln  $FF(\Theta, \Delta, k, l)$  und  $FP(\Theta)$  keine Werte zugeordnet sind.
2. Trivial (da  $FF(\Theta, \Delta, k, l)$  kein Wert zugeordnet ist).
3. Trivial (da  $FF(\Theta, \Delta, k_1, l_1)$  kein Wert zugeordnet ist).

Fall 2:  $\Theta$  hat die Form  $\langle \rangle$ .

1. Durch Punkt 14 von Definition 4.2.5 wird  $FF(\Theta, \Delta, k, l)$  ein eindeutiger Wert zugeordnet, wenn  $l = k$ . Wenn  $l \neq k$ , so wird  $FF(\Theta, \Delta, k, l)$  kein Wert zugeordnet. Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  sind in jedem Falle keine Werte zugeordnet.
2. Wenn  $FF(\Theta, \Delta, k, l)$  definiert ist, ist  $k = l$ .
3. Wenn  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind, dann ist  $l_1 \setminus k_1 = l_2 \setminus k_2 = \emptyset$ .

### Induktionsschritt

Sei  $\Theta$  ein PRS-Gebilde der Komplexität  $m > 0$ . Man nehme an, dass die Induktionsaussage für alle  $n \leq m$  gilt. (In den unten aufgeführten Fällen soll  $\Sigma$  immer eine nicht-leere Folge von PRS-Bedingungen bezeichnen.)

Fall 1:  $\Theta$  hat die Form  $\neg B$ .

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FB(\Theta, \Delta, k)$  höchstens ein Wert zugeordnet wird, nämlich durch Punkt 4 von Definition 4.2.5 ( $\langle B \rangle$  hat eine niedrigere Komplexität als  $\neg B$ , denn  $K(\langle B \rangle) = K(B) + 1$  und  $K(\neg B) = K(B) + 3$ ). Den Formeln  $FF(\Theta, \Delta, k, l)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Trivial.
3. Trivial.

Fall 2:  $\Theta$  hat die Form  $B \Rightarrow C$ .

1. Aus Punkten 1 und 4 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FB(\Theta, \Delta, k)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 5 der Definition). Den Formeln  $FF(\Theta, \Delta, k, l)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Trivial.
3. Trivial.

Fall 3:  $\Theta = \langle \text{math\_id}(d, m) \rangle$ .

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 6 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Dann ist wegen Punkt 6 der Definition  $l = k \cup \{ \langle d, m \rangle \}$  und somit  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Dann ist wegen Punkt 6 der Definition  $l_1 \setminus k_1 = l_2 \setminus k_2 = \{ \langle d, m \rangle \}$ .

Fall 4:  $\Theta = \langle \text{math\_id}(d, m) \rangle + \Sigma$ .

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 7 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Da dies nur über Punkt 7 der Definition definiert sein kann, muss auch  $FF(\Sigma, \Delta, k \cup \{ \langle d, m \rangle \}, l)$  definiert sein. Dann ist wegen Punkt 2 der Induktionsannahme  $k \cup \{ \langle d, m \rangle \} \subseteq l$  und somit  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Weil diese nur über Punkt 7 definiert sein können, müssen auch  $FF(\Sigma, \Delta, k_1 \cup \{ \langle d, m \rangle \}, l_1)$  und  $FF(\Sigma, \Delta, k_2 \cup \{ \langle d, m \rangle \}, l_2)$  definiert sein. Aus Punkt 3 der Induktionsannahme folgt dann  $l_1 \setminus (k_1 \cup \{ \langle d, m \rangle \}) = l_2 \setminus (k_2 \cup \{ \langle d, m \rangle \})$ . Wegen Punkt 2 der Induktionsannahme gilt außerdem  $\langle d, m \rangle \in l_1$  und  $\langle d, m \rangle \in l_2$ , und somit  $l_1 \setminus k_1 = l_2 \setminus k_2$ .

Fall 5:  $\Theta = \langle C := D \rangle$ .

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 8 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Dann ist wegen Punkt 8 der Definition  $l = k$  und somit  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Dann ist wegen Punkt 8 der Definition  $l_1 \setminus k_1 = l_2 \setminus k_2 = \emptyset$ .

Fall 6:  $\Theta = \langle C := D \rangle + \Sigma$ .

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 9 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Da diese Formel nur über Punkt 9 der Definition definiert sein kann, muss eine Formel der Form  $FF(\Sigma, \Delta', k, l)$  definiert sein. Dann ist wegen Punkt 2 der Induktionsannahme  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Weil diese nur über Punkt 9 definiert sein können, müssen Formeln der Form  $FF(\Sigma, \Delta', k_1, l_1)$  und  $FF(\Sigma, \Delta', k_2, l_2)$  definiert sein. Aus Punkt 3 der Induktionsannahme folgt dann  $l_1 \setminus k_1 = l_2 \setminus k_2$ .

Fall 7:  $\Theta = \langle B \rangle$ .

1. Aus den Punkten 1 und 4 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 10 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Dann ist wegen Punkt 10 der Definition  $l = k$  und somit  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Dann ist wegen Punkt 10 der Definition  $l_1 \setminus k_1 = l_2 \setminus k_2 = \emptyset$ .

Fall 8:  $\Theta = \langle B \rangle + \Sigma$ .

1. Aus den Punkten 1 und 4 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 11 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Da diese Formel nur über Punkt 11 der Definition definiert sein kann, muss es ein  $l'$  geben, so dass  $FF(\Sigma, \Delta, k \cup (l' \setminus l), l')$  definiert ist und  $k \cap (l' \setminus l) = \emptyset$  ist. Dann ist wegen Punkt 2 der Induktionsannahme  $k \cup (l' \setminus l) \subseteq l'$ . Daraus folgt, dass  $k \subseteq l$  ist (denn wenn  $x \in k$ , dann ist  $x \notin (l' \setminus l)$  aber  $x \in l'$ , und somit  $x \in l$ ).
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Weil diese nur über Punkt 11 definiert sein können, muss es  $l'_1$  und  $l'_2$  geben, so dass  $FF(\Sigma_B, \Delta, l_1, l'_1)$ ,  $FF(\Sigma_B, \Delta, l_2, l'_2)$ ,  $FF(\Sigma, \Delta, k_1 \cup (l'_1 \setminus l_1), l'_1)$  und  $FF(\Sigma, \Delta, k_2 \cup (l'_2 \setminus l_2), l'_2)$  definiert sind. Aus Punkt 2 der Induktionsannahme folgt  $l_1 \subseteq l'_1$  und  $l_2 \subseteq l'_2$ , und aus Punkt 3 der Induktionsannahme folgt  $l'_1 \setminus (k_1 \cup (l'_1 \setminus l_1)) = l'_2 \setminus (k_2 \cup (l'_2 \setminus l_2))$ .

**Lemma 4.4.2.** Für  $i=1,2$  gilt  $l'_i \setminus (k_i \cup (l'_i \setminus l_i)) = l_i \setminus k_i$

**Beweis:**

„ $\subseteq$ “: Sei  $x \in l'_i \setminus (k_i \cup (l'_i \setminus l_i))$ . Dann ist  $x \in l'_i$  und  $x \notin k_i \cup (l'_i \setminus l_i)$ , also  $x \notin l'_i \setminus l_i$ , also  $x \in l_i$ .  $x \notin k_i$ , also  $x \in l_i \setminus k_i$ .

„ $\supseteq$ “: Sei  $x \in l_i \setminus k_i$ . Dann ist  $x \in l_i$ , also  $x \in l'_i$  und  $x \notin l'_i \setminus l_i$ . Außerdem ist  $x \notin k_i$ , also  $x \notin (k_i \cup (l'_i \setminus l_i))$ , also  $x \in l'_i \setminus (k_i \cup (l'_i \setminus l_i))$ .

□

Folglich ist  $l_1 \setminus k_1 = l_2 \setminus k_2$ .

Fall 9:  $\Theta = \langle c \rangle$  für eine PRS-Bedingung  $c$ , die keine PRS, Definitionsbedingung oder *math\_id*-Bedingung ist.

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 12 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Dann ist wegen Punkt 12 der Definition  $l = k$  und somit  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Dann ist wegen Punkt 12 der Definition  $l_1 \setminus k_1 = l_2 \setminus k_2 = \emptyset$ .

Fall 10:  $\Theta = \langle c \rangle + \Sigma$  für eine PRS-Bedingung  $c$ , die keine PRS, Definitionsbedingung oder *math\_id*-Bedingung ist.

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FF(\Theta, \Delta, k, l)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 13 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FP(\Theta)$  wird kein Wert zugeordnet.
2. Man nehme an, dass  $FF(\Theta, \Delta, k, l)$  definiert ist. Da diese Formel nur über Punkt 13 der Definition definiert sein kann, muss  $FF(\Sigma, \Delta, k, l)$  definiert sein. Dann ist wegen Punkt 2 der Induktionsannahme  $k \subseteq l$ .
3. Man nehme an, dass  $FF(\Theta, \Delta, k_1, l_1)$  und  $FF(\Theta, \Delta, k_2, l_2)$  definiert sind. Weil diese nur über Punkt 13 definiert sein können, müssen  $FF(\Sigma, \Delta, k_1, l_1)$  und  $FF(\Sigma, \Delta, k_2, l_2)$  definiert sein. Aus Punkt 3 der Induktionsannahme folgt dann  $l_1 \setminus k_1 = l_2 \setminus k_2$ .

Fall 11:  $\Theta$  ist eine PRS.

1. Aus Punkt 1 der Induktionsannahme für  $n \leq m$  folgt, dass der Formel  $FP(\Theta)$  höchstens ein Wert zugeordnet wird (nämlich durch Punkt 15 der Definition). Den Formeln  $FB(\Theta, \Delta, k)$  und  $FF(\Theta, \Delta, k, l)$  wird kein Wert zugeordnet.
2. Trivial.
3. Trivial.

Fall 12:  $\Theta$  hat eine Form, die nicht durch Fälle 1-11 abgedeckt ist.

Dann ist kein Formelbild für  $\Theta$  definiert, und die Induktionsannahme ist triviale Weise war. □

Nachdem wir jetzt die Wohldefiniertheit des Formelbildes gezeigt haben, müssen wir noch zeigen, dass jede wohlgeformte PRS ein Formelbild hat, und dass dieses Formelbild ein Satz der Formelsprache ist. Dafür benötigen wir das folgende Lemma:

**Lemma 4.4.3.** *Sei  $B$  eine PRS und  $c_1$  eine PRS-Bedingung. Seien  $\Delta, k, l$  partielle Funktionen.*

*Sei  $P(\Theta, \Delta, k)$  die Aussage, dass*

1. *für jede Sub-PRS  $C$  von  $\Theta$  und jede Bedingung  $c \in \Sigma_C$  der Form  $math\_id(d, m)$   $m \in M_C$  gilt,*

2. für jede Bedingung  $c$  der Form  $\text{math\_id}(d, m)$ , die in einer Sub-PRS von  $\Theta$  vorkommt,  $d \notin \text{Def}(k)$  gilt und es keine von  $c$  aus indirekt zugängliche Bedingung der Form  $\text{math\_id}(d, m')$  gibt,
3. für jede Bedingung  $c$  der Form  $\text{math\_id}(d, m)$  und jede Variable  $x \in \text{frei}(m)$  entweder  $x \in \text{Bild}(k)$  oder es eine von  $c$  aus indirekt zugänglich Bedingung der Form  $\text{math\_id}(d', x)$  gibt,
4. für jedes  $m \in \text{Bild}(k)$   $\text{frei}(m) \subseteq \text{Bild}(k)$  gilt,
5. für jede Bedingung der Form  $\text{holds}(d)$ , die in einer Sub-PRS von  $\Theta$  vorkommt und von der aus keine Bedingung der Form  $\text{math\_id}(d, \varphi)$  für eine Formel  $\varphi$  indirekt zugänglich ist,  $k(d)$  definiert ist und eine Formel ist,
6. für jede Bedingung  $c$  der Form  $p(d_1, \dots, d_n)$ , die in einer Sub-PRS von  $\Theta$  vorkommt,  $k(d_i)$  keine Formel ist und keine Bedingung der Form  $\text{math\_id}(d_i, \varphi)$  von  $c$  aus indirekt zugänglich ist (für  $1 \leq i \leq n$  und für eine Formel  $\varphi$ ), und
7. für jedes  $\varphi \in \text{Bild}(\Delta)$ ,  $\text{frei}(\varphi) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k)$ .

Sei  $Q(B, k, l)$  die Aussage, dass  $k \subseteq l$  und  $l \setminus k = \{\langle d, m \rangle : \text{math\_id}(d, m) \in \Sigma_B\}$ .

1. Wenn  $P(B, \Delta, k)$  und  $Q(B, k, l)$  gelten, dann ist  $FF(\Sigma_B, \Delta, k, l)$  definiert, und

$$\text{frei}(FF(\Sigma_B, \Delta, k, l)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l). \quad (4.4)$$

2. Wenn  $c_1$  keine PRS, Definitionsbedingung oder  $\text{math\_id}$ -Bedingung ist und  $P(c_1, \Delta, k)$  gilt, dann ist  $FB(c_1, \Delta, k)$  definiert, und

$$\text{frei}(FB(c_1, \Delta, k)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k). \quad (4.5)$$

### Beweis

Der Beweis verläuft über eine simultane Induktion über die Komplexität von  $\Sigma_B$  und  $c_1$ .

### Induktionsanfang

Angenommen  $K(\Sigma_B) = K(c_1) = 0$ . Dann ist  $\Sigma_B = \langle \rangle$ , und  $c_1$  ist eine Bedingung der Form  $\text{holds}(d)$ ,  $p(d_1, \dots, d_n)$ ,  $\text{math\_id}(d, m)$  oder  $\text{contradiction}$ . Dann ist  $FF(\Sigma_B, \Delta, k, l)$  wegen Punkt 14 der Formelbilddefinition definiert und gleich  $\top$ . Somit ist  $\text{frei}(FF(\Sigma_B, \Delta, k, l)) = \emptyset$ , woraus 4.4 folgt.

Nun nehme man an, dass  $c_1$  keine PRS, Definitionsbedingung oder  $\text{math\_id}$ -Bedingung ist und  $P(c_1, \Delta, k)$  gilt.

Wenn  $c_1$  die Form  $holds(d)$  hat, dann folgt aus Punkt 5 von  $P(c_1, \Delta, k)$ , dass  $k(d)$  definiert ist und eine Formel  $\varphi$  ist, so dass nach Punkt 1 der Formelbilddefinition  $FB(c_1, \Delta, k)$  definiert und gleich  $\bar{\Delta}(\varphi)$  ist. Da  $\varphi \in Bild(k)$ , folgt aus Punkt 4, dass  $frei(\varphi) \subseteq Bild(k)$ . Aus Punkt 7 und der Definition von  $\bar{\Delta}$  folgt dann, dass  $\bar{\Delta}(\varphi) \subseteq Bild(k)$ , woraus 4.5 folgt..

Wenn  $c_1$  die Form  $p(d_1, \dots, d_n)$  hat, dann folgt aus Punkt 6 von  $P(c_1, \Delta, k)$ , dass  $k(d_i)$  für  $1 \leq i \leq n$  keine Formel ist. Aus Punkt 2 der Formelbilddefinition folgt dann, dass  $FB(c_1, \Delta, k)$  definiert ist und gleich  $\bar{\Delta}(p(\widehat{k}(d_1), \dots, \widehat{k}(d_n)))$  ist. Sei  $x \in frei(FB(c_1, \Delta, k)) \setminus \{\delta_i : i \in \mathbb{N}\}$ . Dann kommt  $x$  in einem  $\widehat{k}(d_i)$  vor (für ein  $i$  mit  $1 \leq i \leq n$ ). Da  $x$  nicht von der Form  $\delta_j$  ist, ist  $\widehat{k}(d_i)$  nicht von der Form  $\delta_j$ , so dass  $\widehat{k}(d_i) = k(d_i)$ . Da  $k(d_i)$  keine Formel ist, ist es ein Term, so dass  $x \in frei(k(d_i))$ . Aus Punkt 4 folgt dann, dass  $x \in Bild(k)$ . Somit gilt 4.5.

Wenn  $c_1$  die Form  $contradiction$  hat, dann folgt aus Punkt 3 der Formelbilddefinition, dass  $FB(c_1, \Delta, k)$  definiert und gleich  $\perp$  ist, womit trivialerweise auch 4.5 gilt.

### Induktionsschritt

Man nehme an, dass  $K(B) = K(c_1) = n > 0$ , und dass die Aussage des Lemmas für alle PRSen und Bedingungen mit einer niedrigeren Komplexität als  $n$  gilt.

1. Angenommen  $P(B, \Delta, k)$  und  $Q(B, k, l)$  gelten. Wir zeigen jetzt für die verschiedenen Formen, die  $\Sigma_B$  annehmen kann, dass  $FF(\Sigma_B, \Delta, k, l)$  definiert ist und 4.4 gilt. (In den unten aufgeführten Fällen soll  $\Sigma$  immer eine nicht-leere Folge von PRS-Bedingungen bezeichnen.)

Fall 1:  $\Sigma_B = \langle math\_id(d, m) \rangle$ .

Aus  $Q(B, k, l)$  folgt, dass  $l = k \cup \{d, m\}$ . Somit folgt aus Punkt 6 der Formelbilddefinition, dass  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $\top$  ist, woraus trivialerweise auch 4.4 folgt.

Fall 2:  $\Sigma_B = \langle math\_id(d, m) \rangle + \Sigma$ .

Sei  $B'$  die PRS, die dieselben Diskursreferenten und mathematischen Referenten wie  $B$  hat, und die  $\Sigma$  als Bedingungsliste hat. Dann folgt aus  $Q(B, k, l)$  und Punkt 2 von  $P(B, \Delta, k)$ , dass  $Q(B', k \cup \{d, m\}, l)$  gilt. Außerdem kann leicht gezeigt werden, dass aus  $P(B, \Delta, k)$   $P(B', \Delta, k \cup \{d, m\})$  folgt. Aus der Induktionsannahme folgt dann, dass  $FF(\Sigma, \Delta, k \cup \{d, m\}, l)$  definiert ist, und dass

$$frei(FF(\Sigma, \Delta, k \cup \{d, m\}, l)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq Bild(l).$$

Somit folgt aus Punkt 7 der Formelbilddefinition, dass  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $FF(\Sigma, \Delta, k \cup \{d, m\}, l)$  ist, woraus wir 4.4 schließen können.



Fall 3:  $\Sigma_B = \langle C := D \rangle$ .

Aus  $Q(B, k, l)$  folgt, dass  $l = k$ . Somit folgt aus Punkt 8 der Formelbilddefinition, dass  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $\top$  ist, woraus trivialerweise auch 4.4 folgt.

Fall 4:  $\Sigma_B = \langle C := D \rangle + \Sigma$ .

Sei  $k' := k \cup \{\langle d, m \rangle : \text{math\_id}(d, m) \in \Sigma_C\}$ . Dann gilt  $Q(C, k, k')$ . Da  $C$  eine Sub-PRS von  $B$  ist, lässt sich leicht zeigen, dass aus  $P(B, \Delta, k)$   $P(C, \Delta, k)$  folgt. Somit folgt aus der Induktionsannahme, dass  $FF(\Sigma_C, \Delta, k, k')$  definiert ist.

Sei  $B_D$  die PRS, die keine Diskursreferenten oder mathematischen Referenten hat, und die nur  $D$  als Bedingung hat. Trivialerweise gilt  $Q(\Delta, k', k')$ . Da  $D$  eine Sub-PRS von  $B$  ist, und die einzigen *math\_id*-Bedingungen, die von Bedingungen in  $D$  aus indirekt zugänglich sind ohne selber in  $D$  zu sein die *math\_id*-Bedingungen in  $C$  sind, deren Zuordnungen in  $k'$  gegeben sind, lässt sich leicht zeigen, dass aus  $P(B, \Delta, k)$   $P(B_D, \Delta, k')$  folgt. Somit folgt aus der Induktionsannahme, dass  $FF(\langle D \rangle, \Delta, k', k')$  definiert ist, und dass

$$\text{frei}(FF(\langle D \rangle, \Delta, k', k')) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k'). \quad (4.6)$$

Aufgrund der Einschränkungen der Definitionsbedingungen in Definition 3.1.2 wissen wir, dass entweder  $C := D$  eine Formel  $R(x_1, \dots, x_m)$  definiert, oder  $C := D$  einen Prädikator-Ausdruck  $p(d_1, \dots, d_m)$  mittels  $\alpha_1, \dots, \alpha_m$  definiert. Im ersten Fall setzen wir  $\mathfrak{P} := R$ . Im zweiten Fall setzen wir  $\mathfrak{P} := p$  und  $x_i := \delta(\alpha_i)$  für  $1 \leq i \leq m$ . In beiden Fällen folgt aus Definition 3.1.2, dass

$$\{m : \text{es gibt ein } d, \text{ so dass } \text{math\_id}(d, m) \in \Sigma_C\} \subseteq \{x_1, \dots, x_m\}.$$

Somit folgt aus 4.6 und der Definition von  $k'$ , dass

$$\text{frei}(\lambda x_n \dots \lambda x_1. FF(\langle D \rangle, \Delta, k', k')) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k). \quad (4.7)$$

Sei  $\Delta' := \Delta \cup \{\langle \mathfrak{P}, \lambda x_n \dots \lambda x_1. FF(\langle D \rangle, \Delta, k', k') \rangle\}$ , und sei  $B'$  die PRS, die dieselben Diskursreferenten und mathematischen Referenten wie  $B$  hat, und die  $\Sigma$  als Bedingungsliste hat. Aus  $Q(B, k, l)$  folgt dann  $Q(B', k, l)$ . Außerdem kann leicht gezeigt werden, dass  $P(B', \Delta', k)$  aus  $P(B, \Delta, k)$  und 4.7 folgt. Aus der Induktionsannahme folgt dann, dass  $FF(\Sigma, \Delta', k, l)$  definiert ist, und dass

$$\text{frei}(FF(\Sigma, \Delta', k, l)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l).$$

Somit folgt aus Punkt 9 der Formelbilddefinition, dass  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $FF(\Sigma, \Delta', k, l)$  ist, womit auch 4.4 gilt.

Fall 5:  $\Sigma_B = \langle C \rangle$ .

Sei  $l' := k \cup \{\langle d, m \rangle : \text{math\_id}(d, m) \in \Sigma_C\}$ . Dann gilt  $Q(C, k, l')$ . Da  $C$  eine Sub-PRS von  $B$  ist, lässt sich leicht zeigen, dass  $P(C, \Delta, k)$  aus  $P(B, \Delta, k)$  folgt.

Somit folgt aus der Induktionsannahme, dass  $FF(\Sigma_C, \Delta, k, l')$  definiert ist, und dass

$$\text{frei}(FF(\Sigma_C, \Delta, k, l')) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l'). \quad (4.8)$$

Somit folgt aus Punkt 10 der Formelbilddefinition, dass  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $\exists X_{C,\nu} FF(\Sigma_C, \Delta, k, l')$  ist.

Sei  $x \in \text{Bild}(l') \setminus \text{Bild}(k)$ . Dann ist eine Bedingung der Form  $\text{math\_id}(d, x)$  in  $\Sigma_C$ . Aus Punkt 1 von  $P(B, \Delta, k)$  folgt damit, dass  $x \in M_C$ . Aus Definition 4.2.3 folgt dann  $x \in Z_C$ , also  $x \in X_{C,\nu}$ . Somit ist  $\text{Bild}(l') \setminus \text{Bild}(k) \subseteq X_{C,\nu}$ . Hiermit können wir 4.4 aus 4.8 schließen.

Fall 6:  $\Sigma_B = \langle C \rangle + \Sigma$ .

Sei  $l' := l \cup \{\langle d, m \rangle : \text{math\_id}(d, m) \in \Sigma_C\}$ . Aus Punkt 2 von  $P(B, \Delta, k)$  folgt, dass jedes  $\langle d, m \rangle$ , für das  $\text{math\_id}(d, m) \in \Sigma_C$  gilt, nicht in  $k$ , und somit auch nicht in  $l$  ist, so dass  $k \cap (l' \setminus l) = \emptyset$ .

Aus der Definition von  $l'$  folgt  $Q(C, l, l')$ . Außerdem kann man zeigen, dass  $P(C, \Delta, l)$  aus  $P(B, \Delta, k)$  folgt. Somit folgt aus der Induktionsannahme, dass  $FF(\Sigma_C, \Delta, l, l')$  definiert ist, und dass

$$\text{frei}(FF(\Sigma_C, \Delta, l, l')) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l'). \quad (4.9)$$

Sei  $B'$  die PRS, die dieselben Diskursreferenten und mathematischen Referenten wie  $B$  hat, und die  $\Sigma$  als Bedingungsliste hat. Da  $k \cap (l' \setminus l) = \emptyset$ , lässt sich leicht zeigen, dass  $l' \setminus (k \cup (l' \setminus l)) = l \setminus k$ , so dass  $Q(B', k \cup (l' \setminus l), l')$  gilt. Da die einzigen  $\text{math\_id}$ -Bedingungen, die von Bedingungen in  $\Sigma$  aus indirekt zugänglich sind ohne selber in  $\Sigma$  zu sein die  $\text{math\_id}$ -Bedingungen in  $C$  sind, deren Zuordnungen in  $l' \setminus l$  gegeben sind, lässt sich leicht zeigen, dass aus  $P(B, \Delta, k)$   $P(B', \Delta, k \cup (l' \setminus l))$  folgt. Somit folgt aus der Induktionsannahme, dass  $FF(\Sigma, \Delta, k \cup (l' \setminus l), l')$  definiert ist, und dass

$$\text{frei}(FF(\Sigma, \Delta, k \cup (l' \setminus l), l')) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l'). \quad (4.10)$$

Somit ist aufgrund von Punkt 11 der Formelbilddefinition  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $\exists X_{C,\nu}(FF(\Sigma_C, \Delta, l, l') \wedge FF(\Sigma, \Delta, k \cup (l' \setminus l), l'))$ . Analog zum letzten Fall lässt sich zeigen, dass  $\text{Bild}(l') \setminus \text{Bild}(k) \subseteq X_{C,\nu}$ . Damit folgt 4.4 aus 4.9 und 4.10.

Fall 7:  $\Sigma_B = \langle c \rangle$  für eine PRS-Bedingung  $c$ , die keine PRS, Definitionsbedingung oder  $\text{math\_id}$ -Bedingung ist.

Es lässt sich leicht zeigen, dass  $P(B', \Delta, k)$  aus  $P(B, \Delta, k)$  folgt. Somit ist aufgrund der Induktionsannahme  $FB(c, \Delta, k)$  definiert, und

$$\text{frei}(FB(c_1, \Delta, k)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k). \quad (4.11)$$

Nach Punkt 11 der Formelbilddefinitionen ist  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $FB(c, \Delta, k)$ . Aus 4.11 folgt dann 4.4.

Fall 8:  $\Sigma_B = \langle c \rangle + \Sigma$  für eine PRS-Bedingung  $c$ , die keine PRS, Definitionsbedingung oder *math\_id*-Bedingung ist.

Sei  $B'$  die PRS, die dieselben Diskursreferenten und mathematischen Referenten wie  $B$  hat, und die  $\Sigma$  als Bedingungsliste hat. Aus  $Q(B, k, l)$  folgt  $Q(B', k, l)$ . Es lässt sich leicht zeigen, dass aus  $P(B, \Delta, k)$   $P(c, \Delta, k)$  und  $P(B', \Delta, k)$  folgen. Somit sind aufgrund der Induktionsannahme  $FB(c, \Delta, k)$  und  $FF(\Sigma, \Delta, k, l)$  definiert, und

$$\text{frei}(FF(\Sigma_B, \Delta, k, l)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l), \quad (4.12)$$

$$\text{frei}(FB(c_1, \Delta, k)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k). \quad (4.13)$$

Nach Punkt 12 der Formelbilddefinitionen ist  $FF(\Sigma_B, \Delta, k, l)$  definiert und gleich  $(FB(c, \Delta, k) \wedge FF(\Sigma, \Delta, k, l))$ . Aus 4.12 und 4.13 folgt dann 4.4.

2. Man nehme an, dass  $c_1$  keine PRS, Definitionsbedingung oder *math\_id*-Bedingung ist, und dass  $P(c_1, \Delta, k)$  gilt. Da  $K(c_1) > 0$ , ist  $c_1$  nicht von der Form *holds*( $d$ ),  $p(d_1, \dots, d_m)$  oder *contradiction*. Somit ist  $c_1$  von der Form  $\neg B$  oder  $B \Rightarrow C$ . Wir zeigen jetzt für diese beiden Fälle, dass  $FB(c_1, \Delta, k)$  definiert ist und 4.5 gilt:

Fall 1:  $c_1 = \neg B$ .

Sei  $B'$  definiert durch  $\Sigma_{B'} := \langle B \rangle$ ,  $D_{B'} := \emptyset$  und  $M_{B'} := \emptyset$ .

Da  $K(\neg B) = K(B) + 3$  und  $K(B') = K(\langle B \rangle) + 1 = K(B) + 2$  ist, ist  $K(B') < K(\neg B)$ , so dass die Induktionsannahme für  $B'$  gilt. Trivialerweise gilt  $Q(B', k, k)$ . Außerdem kann leicht gezeigt werden, dass  $P(B', \Delta, k)$  aus  $P(c_1, \Delta, k)$  folgt. Somit ist  $FF(\langle B \rangle, \Delta, k, k)$  definiert, und

$$\text{frei}(FF(\langle B \rangle, \Delta, k, k)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(k). \quad (4.14)$$

Damit ist nach Punkt 4 der Formelbilddefinition  $FB(c_1, \Delta, k)$  definiert und gleich  $\neg FF(\langle B \rangle, \Delta, k, k)$ . Aus 4.14 folgt dann 4.5.

Fall 2:  $c_1 = B \Rightarrow C$ .

Sei  $C'$  definiert durch  $\Sigma_{C'} := \langle C \rangle$ ,  $D_{C'} := \emptyset$  und  $M_{C'} := \emptyset$ .

Sei  $l := k \cup \{\langle d, m \rangle : \text{math\_id}(d, m) \in \Sigma_B\}$ . Dann gelten  $Q(B, k, l)$  und  $Q(C', l, l)$ . Es kann leicht gezeigt werden, dass  $P(B, \Delta, k)$  aus  $P(c_1, \Delta, k)$  folgt. Da  $C$  eine Sub-PRS von  $c_1$  ist, und die einzigen *math\_id*-Bedingungen, die von Bedingungen in  $C$  aus indirekt zugänglich sind ohne selber in  $C$  zu sein die *math\_id*-Bedingungen in  $B$  sind, deren Zuordnungen in  $l$  gegeben sind, lässt sich außerdem leicht zeigen, dass  $P(C', \Delta, l)$  aus  $P(c_1, \Delta, k)$  folgt. Somit folgt aus der Induktionsannahme, dass  $FF(\Sigma_B, \Delta, k, l)$  und  $FF(\langle C \rangle, \Delta, l, l)$  definiert sind, und dass

$$\text{frei}(FF(\Sigma_B, \Delta, k, l)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l), \quad (4.15)$$

$$\text{frei}(FF(\langle C \rangle, \Delta, l, l)) \setminus \{\delta_i : i \in \mathbb{N}\} \subseteq \text{Bild}(l). \quad (4.16)$$

Aus Punkt 5 der Formelbilddefinition folgt dann, dass  $FB(c_1, \Delta, k)$  definiert und gleich  $\forall X_{B,l}(FF(\Sigma_B, \Delta, k, l) \rightarrow FF(\langle C \rangle, \Delta, l, l))$  ist.

Sei  $x \in \text{Bild}(l) \setminus \text{Bild}(k)$ . Dann ist eine Bedingung der Form  $\text{math\_id}(d, x)$  in  $\Sigma_B$ . Aus Punkt 1 von  $P(c_1, \Delta, k)$  folgt damit, dass  $x \in M_C$ . Aus Definition 4.2.3 folgt dann  $x \in Z_B$ , also  $x \in X_{B,l}$ . Somit ist  $\text{Bild}(l) \setminus \text{Bild}(k) \subseteq X_{B,l}$ . Hiermit können wir 4.5 aus 4.15 und 4.16 schließen.  $\square$

**Theorem 4.4.4.** *Für jede wohlgeformte PRS  $B$  ist  $FP(B)$  definiert und ist ein Satz der Formelsprache.*

### Beweis

Man nehme an, dass  $B$  eine wohlgeformte PRS ist. Sei  $B'$  definiert durch  $\Sigma_{B'} := \langle B \rangle, D_{B'} := \emptyset$  und  $M_{B'} := \emptyset$ .

Dann gilt  $Q(B', \emptyset, \emptyset)$ . Aus der Tatsache, dass  $B$  wohlgeformt ist, folgt außerdem  $P(B', \emptyset, \emptyset, \emptyset)$  (Punkte 4 und 7 von  $P(B', \emptyset, \emptyset, \emptyset)$  sind trivial; Punkte 1, 2, 3, 5 und 6 von  $P(B', \emptyset, \emptyset, \emptyset)$  folgen jeweils aus den Punkten 2, 3, 4, 5 und 6 von Definition 4.0.1). Somit folgt aus Lemma 4.4.3, dass  $FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)$  definiert ist, und dass

$$\text{frei}(FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)) \setminus \{\delta_i : i \in \mathbb{N}\} = \emptyset. \quad (4.17)$$

Aus Punkt 15 der Formelbilddefinition folgt dann, dass  $FB(B)$  definiert und gleich  $FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)$  ist. Um zu beweisen, dass  $FB(B)$  ein Satz ist, muss nur noch gezeigt werden, dass  $FB(B)$  keine freien Variablen der Form  $\delta_i$  enthält:

Die einzige Art, wie eine Variable der Form  $\delta_i$  in dem Formelbild  $FB(B)$  auftauchen kann, ist durch Punkt 2 der Formelbilddefinition. Das heißt, dass ein Schritt der Berechnung des Formelbildes  $FB(B)$  die Berechnung eines Formelbildes der Form  $FB(p(d_1, \dots, d_n), \Delta, k)$  ist, und dass dabei  $d_i \notin \text{Def}(k)$  für ein  $i$  mit  $1 \leq i \leq n$  gilt. In diesem Falle erscheint  $\delta_{d_i}$  in  $FB(p(d_1, \dots, d_n), \Delta, k)$  und damit in  $FB(B)$ . Jetzt muss nur noch gezeigt werden, dass dieses Vorkommnis von  $\delta_{d_i}$  gebunden ist.

Aus Punkt 1 von Definition 4.0.1 folgt, dass  $d_i$  von  $p(d_1, \dots, d_n)$  aus zugänglich ist, also dass es eine von  $p(d_1, \dots, d_n)$  aus zugängliche PRS  $C$  mit  $d_i \in D_C$  gibt. Da  $C$  eine Sub-PRS von  $B$  ist, ist entweder die Berechnung eines Formelbildes der Form  $FF(\langle C \rangle, \Delta', l, l)$  oder die Berechnung eines Formelbildes der Form  $FF(\langle C \rangle + \Sigma, \Delta', k', l)$  (für eine nicht-leere Folge  $\Sigma$ ) zur Berechnung von  $FB(B)$  notwendig. Im ersten Falle folgt aus Punkt 10 der Formelbilddefinition, dass es ein  $l'$  gibt, so dass  $FF(\langle C \rangle, \Delta', l, l) = \exists X_{C,l'} FF(\Sigma_C, \Delta', l, l')$ . Im zweiten Fall folgt aus Punkt 11 der Formelbilddefinition, dass  $FF(\langle C \rangle + \Sigma, \Delta', k', l) = \exists X_{C,l'} (FF(\Sigma_C, \Delta', l, l') \wedge FF(\Sigma, \Delta', k' \cup (l' \setminus l), l'))$ .

Da  $p(d_1, \dots, d_n) \in \Sigma_C$  ist, ist in beiden Fällen die Berechnung von  $FB(p(d_1, \dots, d_n), \Delta, k)$  Teil der Berechnung von  $FF(\Sigma_C, \Delta', l, l')$ , womit auch das oben erwähnte Vorkommnis der Variable  $\delta_{d_i}$  im Skopus von  $\exists X_{C,l'}$  ist. Bei

der Berechnung eines Formelbildes wird die Funktion, die die Informationen der *math\_id*-Bedingungen speichert, nur erweitert oder konstant gelassen, aber nie beschränkt. Somit gilt  $k \subseteq l$ , und hiermit  $d_i \notin Def(l)$ . Da  $d_i \in D_C$ , folgt aus Definition 4.2.3, dass  $\delta_{d_i} \in Y_{C,l} \subseteq X_{C,l}$ . Somit ist das erwähnte Vorkommen von  $\delta_{d_i}$  gebunden.  $\square$



## Kapitel 5

---

# Äquivalenz einer PRS und ihres Formelbildes

---

Nachdem wir nun gezeigt haben, dass für jede wohlgeformte PRS ein eindeutiges Formelbild definiert ist, zeigen wir in diesem Kapitel, dass jede wohlgeformte PRS zu ihrem Formelbild äquivalent ist, und somit das Formelbild eine *korrekte* Übersetzung der PRS ist. Äquivalenz heißt hier im Wesentlichen, dass die PRS und ihr Formelbild von genau denselben Strukturen erfüllt werden. Die genaue Formulierung von dieser Äquivalenzbeziehung findet sich in Theorem 5.2.2.

### 5.1 Hilfskonstruktionen

Für den Beweis dieser Äquivalenz brauchen wir einige Hilfskonstruktionen, die in diesem Abschnitt eingeführt werden.

Wie im vorigen Kapitel bezeichnen auch in diesem Kapitel  $k$  und  $l$  Funktionen, die die Informationen aus *math\_id*-Beziehungen speichern, also Funktionen, die Diskursreferenten Formeln und Terme zuordnen. Die Funktionen  $i, j_0, \dots, j_l$ , die in der Definition der PRS-Semantik (3.2.9) verwendet werden, ordnen den Diskursreferenten allerdings nicht Formeln und Terme zu, sondern Formeln und Objekte des Universums. Wenn eine Teilbelegung  $\sigma$  gegeben ist, können wir jeder Funktion der ersten Art eine Funktion der zweiten Art zuordnen:

**Definition 5.1.1.** Sei  $d$  eine Diskursreferent und  $k$  eine Funktion von Diskursreferenten nach  $L$ -Formeln und  $L$ -Termen. Sei  $\sigma$  eine Teilbelegung.

$$k_\sigma(d) := \begin{cases} k(d) & \text{wenn } k(d) \text{ eine Formel ist} \\ \bar{\sigma}(k(d)) & \text{wenn } k(d) \text{ ein Term ist} \end{cases}$$

$k_\sigma$  ordnet wie  $i, j_0, \dots, j_l$  aus 3.2.9 den Diskursreferenten Formeln und Objekte des Universums zu.

Ein Formelbild einer PRS-Bedingung oder einer Folge von PRS-Bedingungen kann freie Variablen enthalten. Wenn wir die Korrektheit dieser Formelbilder überprüfen, reicht es also nicht aus, Strukturen in Betracht zu ziehen, sondern es müssen auch Teilbelegungen für die freien Variablen des Formelbildes betrachtet werden. Dabei ist es sinnvoll, zwischen den Variablen von  $L$  und den Variablen der Form  $\delta_i$  zu unterscheiden. Die Teilbelegung auf den Variablen von  $L$  bezeichnen wir mit  $\sigma$ ; die Teilbelegung auf den Variablen der Form  $\delta_i$  bezeichnen wir mit  $\iota$ ; somit bezeichnet  $\sigma \cup \iota$  die Teilbelegung auf allen Variablen des Formelbildes.

In Punkt 2 der Formelbilddefinition (4.2.5) wird die Erweiterung  $\widehat{k}$  von  $k$  verwendet, die jedem Diskursreferenten  $d$ , für den  $k(d)$  nicht definiert ist,  $\delta_d$  zuordnet. Mittels einer Teilbelegung  $\iota$ , die den Variablen der Form  $\delta_d$  für  $d \notin \text{Def}(k)$  Elemente des Universums zuordnet, können wir eine Verbindung zwischen  $\widehat{k}$  und  $k_\sigma$  herstellen:

**Lemma 5.1.1.** *Sei  $d$  ein Diskursreferent, sei  $k$  eine Funktion von Diskursreferenten nach  $L$ -Termen und  $L$ -Formeln, und sei  $\iota$  eine Funktion, so dass  $\text{Def}(\iota) \subset \{\delta_n : n \in \mathbb{N}\}$ , und so dass entweder  $k(d)$  ein  $L$ -Term ist und  $\delta_d \notin \text{Def}(\iota)$ , oder  $d \notin \text{Def}(k)$  und  $\delta_d \in \text{Def}(\iota)$ . Sei  $\sigma$  eine Teilbelegung auf den Variablen, die in Termen in  $\text{Bild}(k)$  vorkommen. Dann gilt  $\overline{\sigma \cup \iota}(\widehat{k}(d)) = (k_\sigma \cup (\iota \circ \delta))(d)$  (und beide Seiten der Gleichung sind wohldefiniert).*

**Beweis:**

Fall 1:  $k(d)$  ist ein  $L$ -Term  $t$  und  $\delta_d \notin \text{Def}(\iota)$ . Dann ist  $\widehat{k}(d) = t$  und  $\overline{\sigma \cup \iota}(\widehat{k}(d)) = \overline{\sigma}(t)$  (dies ist wohldefiniert, da  $\iota$  nur auf Variablen der Form  $\delta_n$  definiert ist, und solche Variablen nicht im  $L$ -Term  $t$  vorkommen können).  $\delta(d) = \delta_d$  und  $\iota$  ist nicht auf  $\delta(d)$  definiert. Somit ist  $(\iota \circ \delta)$  nicht auf  $d$  definiert. Daher ist  $(k_\sigma \cup (\iota \circ \delta))(d)$  wohldefiniert und gleich  $k_\sigma(d) = \overline{\sigma}(k(d)) = \overline{\sigma}(t)$ .

Fall 2:  $d \notin \text{Def}(k)$  und  $\delta_d \in \text{Def}(\iota)$ . Da  $k(d)$  nicht definiert ist, ist  $\widehat{k}(d) = \delta_d$ . Da  $\sigma$  auf  $\delta_d$  nicht definiert ist, ist  $\overline{\sigma \cup \iota}(\delta_d) = (\sigma \cup \iota)(\delta_d)$  wohldefiniert und gleich  $\iota(\delta_d)$ . Aus  $d \notin \text{Def}(k)$  folgt auch, dass  $k_\sigma$  nicht auf  $d$  definiert ist, so dass  $(k_\sigma \cup (\iota \circ \delta))(d)$  wohldefiniert und gleich  $(\iota \circ \delta)(d) = \iota(\delta_d)$  ist.  $\square$

**Definition 5.1.2.** Am Anfang von Abschnitt 4.2 wurde  $L'$  als eine Erweiterung von  $L$  definiert, die für jeden Prädikator in  $\Lambda$  um ein neues Relationszeichen erweitert wurde. Dabei haben wir der Einfachheit halber die neuen Relationszeichen von  $L'$  mit den Prädikatoren in  $\Lambda$  gleichgesetzt. Dies bedeutet, dass jede  $(L, \Lambda)$ -Struktur  $\mathfrak{A} = \langle U, \beta \rangle$  auch als  $L'$ -Struktur angesehen werden kann. Um klar zu machen, dass wir  $\mathfrak{A}$  als  $L'$ -Struktur und nicht als  $(L, \Lambda)$ -Struktur meinen, schreiben wir  $\mathfrak{A}'$  anstatt  $\mathfrak{A}$ .



**Definition 5.1.3.** Sei  $\Theta$  ein PRS-Gebilde. Dann ist  $D(\Theta)$  die Menge aller Diskursreferenten, die von Sub-PRSen von  $\Theta$  eingeführt werden, also

$$D(\Theta) := \bigcup \{D_B : B \text{ ist eine Sub-PRS von } \Theta\}.$$

**Definition 5.1.4.** Sei  $\mathfrak{A} = \langle U, \beta \rangle$  eine  $(L, \Lambda)$ -Struktur. Sei  $R$  ein  $n$ -stelliges Relationszeichen, das kein Relationszeichen von  $L$  ist, oder ein  $n$ -stelliger Prädikator, der nicht in  $\Lambda$  ist. Sei  $A \subseteq U^n$ . Dann definieren wir

$$\mathfrak{A} \rtimes \langle R, A \rangle := \langle U, \beta \cup \langle R, A \rangle \rangle.$$

Häufig müssen wir uns auf die Variablen der Form  $\delta_i$  in einem Formelbild beziehen, was uns die folgende Definition ermöglicht:

**Definition 5.1.5.** Sei  $k$  eine Funktion von Diskursreferenten nach Termen, und sei  $\Delta$  eine Funktion von Relationszeichen nach Lambda-Ausdrücken.

Sei  $c$  eine PRS-Bedingung. Dann bezeichnet  $\xi(c, k, \Delta)$  die Menge aller freien Variablen der Form  $\delta_n$  in  $FB(c, \Delta, k)$ .

Sei  $\Theta$  eine nicht-leere Folge von PRS-Bedingungen. Sei  $l$  die Funktion, für die  $FF(\Theta, \Delta, l, k)$  definiert ist (dass  $l$  eindeutig ist, lässt sich leicht aus den Punkten 2 und 3 von Lemma 4.4.1 ableiten). Dann bezeichnet  $\xi(\Theta, k, \Delta)$  die Menge aller freien Variablen der Form  $\delta_n$  in  $FF(\Theta, \Delta, l, k)$ .

## 5.2 Äquivalenzbeweis

In Lemma 5.2.1 zeigen wir die Äquivalenz von PRS-Bedingungen und Folgen von PRS-Bedingungen zu ihren Formelbildern. Daraus lässt sich dann leicht Theorem 5.2.2 ableiten, in dem die Äquivalenz zwischen PRSen und ihren Formelbildern gezeigt wird.

**Lemma 5.2.1.** *Sei  $\Theta$  entweder eine wohlgeformte PRS-Bedingung, die nicht PRS, Definitionsbedingung oder math\_id-Bedingung ist, oder eine wohlgeformte Folge von PRS-Bedingungen.*

*Seien  $k$  und  $l$  partielle Funktionen von Diskursreferenten nach  $L$ -Termen und  $L$ -Formeln, so dass  $Def(k) \cap D(\Theta) = \emptyset$  und  $FB(\Theta, \Delta, k)$  oder  $FF(\Theta, \Delta, l, k)$  definiert ist.*

*Sei  $\mathfrak{A}$  eine  $(L, \Lambda)$ -Struktur und  $\sigma$  eine Teilbelegung auf den Variablen, die in Termen in  $Bild(k)$  vorkommen.*

*Sei  $\Delta$  eine partielle Funktion, die einigen der Relationszeichen und Prädikatoren von  $\mathfrak{A}$  Lambda-Ausdrücke zuordnet.*

*Sei  $\iota$  eine Funktion, so dass*

- $\xi(\Theta, k, \Delta) \subseteq Def(\iota) \subset \{\delta_n : n \in \mathbb{N}\}$ ,
- $Def(\iota \circ \delta) \cap (Def(k) \cup D(\Theta)) = \emptyset$ ,

- $Bild(\iota) \subseteq |\mathfrak{A}|$ .

Es sei gegeben, dass für alle  $R \in Def(\Delta)$

$$\mathfrak{A}', \sigma \cup \iota \models \forall x_1, \dots, x_{|R|} (R(x_1, \dots, x_{|R|}) \leftrightarrow \Delta(R)(x_1) \dots (x_{|R|})). \quad (5.1)$$

1. Wenn  $\Theta$  eine PRS-Bedingung ist, die nicht PRS, Definitionsbedingung oder *math\_id*-Bedingung ist, dann gilt

$$\mathfrak{A}', \sigma \cup \iota \models FB(\Theta, \Delta, k) \Leftrightarrow [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma] \models \Theta.$$

2. Wenn  $\Theta$  eine Folge von PRS-Bedingungen der Länge  $n$  ist, dann gilt  $\mathfrak{A}', \sigma \cup \iota \models FF(\Theta, \Delta, l, k)$  genau dann, wenn es  $\mathfrak{A}_1, \dots, \mathfrak{A}_n, j_1, \dots, j_n, \tau_1, \dots, \tau_n$  gibt, so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q)$  für  $1 \leq q \leq n$ , wobei  $\mathfrak{A}_0 := \mathfrak{A}, \tau_0 := \sigma$  und  $j_0 := k_\sigma \cup (\iota \circ \delta)$ .

### Beweis

Der Beweis verfolgt über eine Induktion über die Komplexität von  $\Theta$ . Man nehme an, dass  $\Theta, k, l, \mathfrak{A}, \sigma, \Delta$  und  $\iota$  gegeben sind und die Annahmen des Lemmas erfüllen. (5.2)

### Induktionsanfang

$\Theta = holds(d)$ :

$$\mathfrak{A}', \sigma \cup \iota \models FB(holds(d), \Delta, k)$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models \overline{\Delta}(k(d))$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models k(d) \text{ (wegen 5.1)}$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models (k_\sigma \cup (\iota \circ \delta))(d), \text{ da } k(d) = k_\sigma(d) = (k_\sigma \cup (\iota \circ \delta))(d) \text{ (weil } k(d) \text{ eine Formel ist, da } FB(holds(d), \Delta, k) \text{ definiert ist)}$$

$$\Leftrightarrow \mathfrak{A}', \sigma \models (k_\sigma \cup (\iota \circ \delta))(d) \text{ (da } \iota \text{ nur auf Variablen der Form } \delta_n \text{ definiert ist, und } (k_\sigma \cup (\iota \circ \delta))(d) = k(d) \text{ eine Formel von } L \text{ ist, und somit keine Variable der Form } \delta_n \text{ enthält)}$$

$$\Leftrightarrow [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma] \models holds(d) \text{ (Punkt 4 von Definition 3.2.9)}$$

$\Theta = p(d_1, \dots, d_n)$ :

$$\mathfrak{A}', \sigma \cup \iota \models FB(p(d_1, \dots, d_n), \Delta, k)$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models \overline{\Delta}(p(\widehat{k}(d_1), \dots, \widehat{k}(d_n)))$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models p(\widehat{k}(d_1), \dots, \widehat{k}(d_n)) \text{ wegen 5.1}$$

$$\Leftrightarrow \langle \overline{\sigma \cup \iota}(\widehat{k}(d_1)) \dots, \overline{\sigma \cup \iota}(\widehat{k}(d_n)) \rangle \in \mathfrak{A}'(p)$$

$$\Leftrightarrow \langle \overline{\sigma \cup \iota}(\widehat{k}(d_1)) \dots, \overline{\sigma \cup \iota}(\widehat{k}(d_n)) \rangle \in \mathfrak{A}(p) \text{ wegen Definition 5.1.2}$$

$$\Leftrightarrow \langle k_\sigma \cup (\iota \circ \delta)(d_1), \dots, k_\sigma \cup (\iota \circ \delta)(d_n) \rangle \in \mathfrak{A}(p) \text{ wegen Lemma 5.1.1 (die Bedingung von diesem Lemma ist erfüllt, da } Def(\iota) \subset \{\delta_n : n \in \mathbb{N}\} \text{ und da aus } \xi(\Theta, k, \Delta) \subseteq Def(\iota) \text{ und } Def(\iota \circ \delta) \cap Def(k) = \emptyset \text{ folgt, dass für alle } d \in \{d_1, \dots, d_n\} \iota \text{ genau dann auf } \delta_d \text{ definiert ist, wenn } k \text{ nicht auf } d \text{ definiert ist)}$$

$$\Leftrightarrow [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma] \models p(d_1, \dots, d_n)$$

$\Theta = \text{contradiction}$ : Trivial.

### Induktionsschritt

$\Sigma$  bezeichnet bei jedem der Fälle eine Folge von PRS-Bedingungen der Länge  $n > 0$ .

$\Theta = \neg B$ :

$$\mathfrak{A}', \sigma \cup \iota \models FB(\neg B, \Delta, k)$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \not\models FF(\langle B \rangle, \Delta, k, k)$$

$\Leftrightarrow$  es gibt keine  $\mathfrak{B}, j, \tau$ , so dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j, \tau] \models B$  (Punkt 2 der Induktionsannahme)

$\Leftrightarrow [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma] \models \neg B$  (Punkt 5 von Definition 3.2.9; für „ $\Leftarrow$ “ beachte man, dass  $\mathfrak{B}$  sowieso gleich  $\mathfrak{A}$  sein muss, um  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j, \tau] \models B$  zu erfüllen).

$\Theta = B \Rightarrow C$ :

$$\mathfrak{A}', \sigma \cup \iota \models FB(B \Rightarrow C, \Delta, k)$$

$\Leftrightarrow$  es gibt ein  $l$ , so dass  $\mathfrak{A}', \sigma \cup \iota \models \forall X_{B,l}(FF(\Sigma_B, \Delta, k, l) \rightarrow FF(\langle C \rangle, \Delta, l, l))$

$\Leftrightarrow$  es gibt ein  $l$ , so dass für jedes  $\tau \supseteq \sigma \cup \iota$  mit  $Def(\tau) = Def(\sigma \cup \iota) \cup X_{B,l}$   $\mathfrak{A}', \tau \models (FF(\Sigma_B, \Delta, k, l) \rightarrow FF(\langle C \rangle, \Delta, l, l))$

$\Leftrightarrow$  es gibt ein  $l$ , so dass  $FF(\Sigma_B, \Delta, k, l)$  definiert ist, und so dass für jedes  $\eta : Y_{B,l} \rightarrow |\mathfrak{A}|$  und jedes  $\tau' \supseteq \sigma$ , für welche  $Def(\tau') = Def(\sigma) \cup Z_B$  und  $\mathfrak{A}', \tau' \cup (\iota \cup \eta) \models FF(\Sigma_B, \Delta, k, l)$  gilt,  $\mathfrak{A}', \tau' \cup (\iota \cup \eta) \models FF(\langle C \rangle, \Delta, l, l)$  (hier wurde das  $\tau$  vom vorherigen Schritt von seinem  $\iota$ -Teil bereinigt und dann in ein  $\tau'$  und ein  $\eta$  aufgeteilt).

$\Leftrightarrow$  es gibt ein  $l$ , so dass  $FF(\Sigma_B, \Delta, k, l)$  definiert ist, und so dass es für jedes  $\eta : Y_{B,l} \rightarrow |\mathfrak{A}|$  und jedes  $\tau' \supseteq \sigma$  mit den Eigenschaften

- $Def(\tau') = Def(\sigma) \cup Z_B$
- es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_m, j_1, \dots, j_m, \tau_1, \dots, \tau_m$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma_B(q)$  für  $1 \leq q \leq m$  (wobei  $n := \text{Länge}(\Sigma_B)$  und  $\mathfrak{A}_0 := \mathfrak{A}, \tau_0 := \tau'$  und  $j_0 := l_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ )

$\tau^*$  und  $j^*$  gibt, so dass  $[\mathfrak{A}, l_{\tau'} \cup ((\iota \cup \eta) \circ \delta), \tau'], [\mathfrak{A}, j^*, \tau^*] \models C$ . Hierbei wurde Punkt 2 der Induktionsannahme zweimal angewendet, einmal für „ $\Theta = \Sigma_B$ “ und einmal für „ $\Theta = \langle C \rangle$ “. Für die erste Anwendung der Induktionsannahme müssen wir zeigen, dass

1.  $\xi(\Sigma_B, l, \Delta) \subseteq Def(\iota \cup \eta) \subset \{\delta_n : n \in \mathbb{N}\}$ ,
2.  $Def((\iota \cup \eta) \circ \delta) \cap (Def(l) \cup D(\Sigma_B)) = \emptyset$ ,

3.  $Bild(\iota \cup \eta) \subseteq |\mathfrak{A}|$ .

**Beweis:**

1. Wegen 5.2 können wir annehmen, dass  $\xi(B \Rightarrow C, k, \Delta) \subseteq Def(\iota) \subseteq \{\delta_n : n \in \mathbb{N}\}$ .

Man nehme an, dass  $\delta_i \in \xi(\Sigma_B, l, \Delta)$  ist. Dann ist  $\delta_i \in frei(FF(\Sigma_B, \Delta, k, l))$ . Da  $FB(B \Rightarrow C, \Delta, k)$  gleich  $\forall X_{B,l} (FF(\Sigma_B, \Delta, k, l) \rightarrow FF(\langle C \rangle, \Delta, l, l))$  ist, ist  $\delta_i \in frei(FB(B \Rightarrow C, \Delta, k))$  oder  $\delta_i \in X_{B,l}$ . Im ersten Falle ist  $\delta_i \in \xi(B \Rightarrow C, k, \Delta) \subseteq Def(\iota)$ , und somit  $\delta_i \in Def(\iota \cup \eta)$ . Im zweiten Falle ist aufgrund von Definition 4.2.3  $\delta_i \in Y_{B,l}$  (da  $Z_B$  keine Variablen der Form  $\delta_i$  enthält), und somit  $\delta_i \in Def(\eta) \subseteq Def(\iota \cup \eta)$ . Damit gilt  $\xi(\Sigma_B, l, \Delta) \subseteq Def(\iota \cup \eta)$ .

Da  $Def(\eta) = Y_{B,l} \subseteq \{\delta_n : n \in \mathbb{N}\}$ , ist  $Def(\iota \cup \eta) = Def(\iota) \cup Def(\eta) \subseteq \{\delta_n : n \in \mathbb{N}\}$ .

2. Aufgrund von 5.2 gilt

$$Def(\iota \circ \delta) \cap (Def(k) \cup D(B \Rightarrow C)) = \emptyset. \quad (5.3)$$

Man nehme an, dass  $d \in Def((\iota \cup \eta) \circ \delta) \cap (Def(l) \cup D(\Sigma_B)) = \emptyset$ . Dann gibt es vier Möglichkeiten:

**Fall 1:**  $d \in Def(\iota \circ \delta)$  und  $d \in Def(l)$ .

In diesem Falle folgt aus 5.3, dass  $d \notin Def(k)$ . Somit ist  $d \in Def(l) \setminus Def(k) \subseteq D_B \subseteq D(B \Rightarrow C)$ , was 5.3 widerspricht.

**Fall 2:**  $d \in Def(\iota \circ \delta)$  und  $D(\Sigma_B)$ .

Da  $D(\Sigma_B) \subseteq D(B \Rightarrow C)$ , ist dies ein Widerspruch zu 5.3.

**Fall 3:**  $d \in Def(\eta \circ \delta)$  und  $d \in Def(l)$ .

$Def(\eta) = Y_{B,l} = \{\delta_d : d \in D_B \setminus Def(l)\}$ , womit  $Def(\eta \circ \delta) = D_B \setminus Def(l)$ . Widerspruch.

**Fall 4:**  $d \in Def(\eta \circ \delta)$  und  $D(\Sigma_B)$ .

Da  $Def(\eta \circ \delta) = D_B \setminus Def(l)$ , ist  $d \in D_B$ . Da  $d \in D(\Sigma_B)$ , ist  $d \in D_C$  für eine Sub-PRS  $C$  von  $\Sigma_B$ . Da  $B$  von  $C$  aus zugänglich ist und  $d \in D_B$ , ist  $d$  von  $C$  aus zugänglich, was Punkt 7 von 4.0.1 widerspricht.

Da jeder der vier Fälle zu einem Widerspruch geführt hat, ist  $Def((\iota \cup \eta) \circ \delta) \cap (Def(l) \cup D(\Sigma_B)) = \emptyset$ .

3. Trivial. □

Für die zweite Anwendung der Induktionsannahme müssen wir zeigen, dass

1.  $\xi(\langle C \rangle, l, \Delta) \subseteq Def(\iota \cup \eta) \subseteq \{\delta_n : n \in \mathbb{N}\}$ ,

2.  $Def((\iota \cup \eta) \circ \delta) \cap (Def(l) \cup D(\langle C \rangle)) = \emptyset$ ,
3.  $Bild(\iota \cup \eta) \subseteq |\mathfrak{A}|$ .

Dies lässt sich auf ähnliche Weise zeigen wie die analoge Aussage bei der ersten Anwendung der Induktionsannahme.

$\Leftrightarrow$  es gibt ein  $l$ , so dass  $FF(\Sigma_B, \Delta, k, l)$  definiert ist, und so dass es für jedes  $\eta : Y_{B,l} \rightarrow |\mathfrak{A}|$  und jedes  $\tau' \supseteq \sigma$ , für welche  $Def(\tau') = Def(\sigma) \cup Z_B$  ist und  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt,  $\tau^*$  und  $j^*$  gibt, so dass  $[\mathfrak{A}, j_0, \tau'], [\mathfrak{A}, j^* \cdot \tau^*] \models C$  (wobei  $j_0 := l_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ ); für diesen Schritt wurde Punkt 1 von Definition 3.2.9 angewendet: es kann leicht gezeigt werden, dass  $k_\sigma \cup (\iota \circ \delta)[d_1, \dots, d_{n_1}]j_0$  und  $\sigma[v_1, \dots, v_{n_2}]\tau'$ , wobei  $d_1, \dots, d_{n_1}$  die Diskursreferenten von  $B$  sind, und  $v_1, \dots, v_{n_2}$  die Variablenreferenten von  $B$  sind).

$\Leftrightarrow$  für alle  $j_0$  und  $\tau'$ , für die  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt, gibt es  $\tau^*$  und  $j^*$ , so dass  $[\mathfrak{A}, j_0, \tau'], [\mathfrak{A}, j^* \cdot \tau^*] \models C$

Für diesen letzten Schritt gelten die beiden Richtungen des Bikonditionals aus den folgenden Gründen:

„ $\Leftarrow$ “: Es gibt immer ein  $l$ , so dass  $FF(\Sigma_B, \Delta, k, l)$  definiert ist.

„ $\Rightarrow$ “: Für alle  $\tau' > \sigma, j_0 > k_\sigma \cup (\iota \circ \delta)$ , so dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt, ist  $Def(\tau') = Def(\sigma) \cup Z_B$  und gibt es ein  $\eta : Y_{B,l} \rightarrow |\mathfrak{A}|$ , so dass  $j_0$  die Form  $l_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$  hat ( $l$  ordnet jedem Diskursreferent  $d$  von  $B$ , für das es ein Term  $t$  gibt, so dass  $math\_id(d, t)$  eine Bedingung in  $B$  ist, den Term  $t$  zu.  $j_0$  muss jedem derartigen  $d$   $\tau'(t)$  zuordnen, um die  $math\_id$ -Bedingungen in  $B$  zu erfüllen. Somit gilt  $l_{\tau'} \subseteq j_0$ .)

$\Theta = \langle math\_id(d, m) \rangle$ :

$FF(\langle math\_id(d, m) \rangle, \Delta, l, k) = \top$ , so dass  $\mathfrak{A}', \sigma \cup \iota \models FF(\langle math\_id(d, m) \rangle, \Delta, l, k)$  in jedem Falle gilt. Da  $FF(\langle math\_id(d, m) \rangle, \Delta, l, k)$  definiert ist, ist  $k = l \cup \{d, m\}$ .

Da  $\delta_d \notin \xi(\Theta, k, \Delta)$  ist, ist  $(k_\sigma \cup (\iota \circ \delta))(d) = k_\sigma(d) = \begin{cases} m & \text{wenn } m \text{ eine Formel ist,} \\ \bar{\sigma}(m) & \text{wenn } m \text{ ein Term ist,} \end{cases}$   
d.h.  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma] \models math\_id(d, m)$ . Somit gilt auch in jedem Falle, dass es  $\mathfrak{A}_1, j_1$  und  $\tau_1$  gibt, so dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}_1, j_1, \tau_1] \models \Theta(1)$ .

$\Theta = \langle math\_id(d, m) \rangle + \Sigma$ :

$\mathfrak{A}', \sigma \cup \iota \models FF(\langle math\_id(d, m) \rangle + \Sigma, \Delta, l, k)$

$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models FF(\Sigma, \Delta, l \cup \{d, m\}, k)$

$\Leftrightarrow$  es gibt  $\mathfrak{A}_2, \dots, \mathfrak{A}_{n+1}, j_2, \dots, j_{n+1}, \tau_2, \dots, \tau_{n+1}$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q-1)$  für  $2 \leq q \leq n+1$ , wobei  $\mathfrak{A}_1 := \mathfrak{A}, \tau_1 := \sigma, j_1 := k_\sigma \cup (\iota \circ \delta)$  (per Induktionsannahme)

$\Leftrightarrow$  es gibt  $\mathfrak{A}_2, \dots, \mathfrak{A}_{n+1}, j_2, \dots, j_{n+1}, \tau_2, \dots, \tau_{n+1}$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q-1)$  für  $2 \leq q \leq n+1$  ( $\mathfrak{A}_1 := \mathfrak{A}, \tau_1 := \sigma, j_1 := k_\sigma \cup (\iota \circ \delta)$ ), und so dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma] \models \text{math\_id}(d, m)$  (denn  $\langle d, m \rangle \in l \cup \{\langle d, m \rangle\} \subseteq k$ )

$\Leftrightarrow$  es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_{n+1}, j_1, \dots, j_{n+1}, \tau_1, \dots, \tau_{n+1}$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q)$  für alle  $1 \leq q \leq n+1$ , wobei  $\mathfrak{A}_0 := \mathfrak{A}, \tau_0 := \sigma, j_0 := k_\sigma \cup (\iota \circ \delta)$  („ $\Rightarrow$ “ trivial; „ $\Leftarrow$ “ gilt, weil eine *math\_id*-Bedingung nie den Kontext ändert, und somit  $[\mathfrak{A}_0, j_0, \tau_0] = [\mathfrak{A}_1, j_1, \tau_1]$  ist).

$\Theta = \langle C := D \rangle$  und  $C := D$  definiert  $R(v_1, \dots, v_m)$  mittels  $d_1, \dots, d_m, d$ :  
 $\mathfrak{A}', \sigma \cup \iota \models FF(\langle C := D \rangle, \Delta, l, k)$  gilt in jedem Falle. Außerdem gilt

$$[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma] \models C := D$$

wobei  $\mathfrak{B}$  als

$$\mathfrak{A} \bowtie \langle R, \{(x_1, \dots, x_m) \in |\mathfrak{A}|^m : \text{es gibt } j_2, \tau_2, \text{ so dass } [\mathfrak{A}, j_{(x_1, \dots, x_m)}, \tau_{(x_1, \dots, x_m)}], [\mathfrak{A}, j_2, \tau_2] \models D\} \rangle$$

definiert wird,  $j_{(x_1, \dots, x_m)}$  eine Erweiterung von  $k_\sigma \cup (\iota \circ \delta)$  auf die Diskursreferenten von  $C$  ist, so dass  $j_{(x_1, \dots, x_m)}(d) := R(v_1, \dots, v_m)$  und  $j_{(x_1, \dots, x_m)}(d_i) := x_i$  für  $1 \leq i \leq n$ , und  $\tau_{(x_1, \dots, x_m)}$  eine Erweiterung von  $\sigma$  auf die Variablenreferenten von  $C$  ist, so dass  $\tau_{(x_1, \dots, x_m)}(v_i) := x_i$  für  $1 \leq i \leq m$ .

### Beweis:

Betrachte Punkt 7 von Definition 3.2.9.  $\mathfrak{B}$  ist eine Erweiterung von  $\mathfrak{A}$  auf das in  $C := D$  definierte Zeichen. Seien  $j_1$  und  $\tau_1$  Funktionen, für die

- $(k_\sigma \cup (\iota \circ \delta))[d, d_1, \dots, d_m]j_1$
- $\sigma[v_1, \dots, v_m]\tau_1$
- $[\mathfrak{B}, j_1, \tau_1], [\mathfrak{B}, j_1, \tau_1] \models c'$  für jede *math\_id*-Bedingung  $c'$  in  $C$

gilt. Zu zeigen ist nun folgendes:  $[\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$  genau dann, wenn es  $j_2, \tau_2$  gibt, so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$ .

„ $\Rightarrow$ “: Angenommen  $[\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$ . Dann gilt  $[\mathfrak{B}, j_1, \tau_1], [\mathfrak{B}, j_1, \tau_1] \models \text{math\_id}(d, R(v_1, \dots, v_m))$  und  $[\mathfrak{B}, j_1, \tau_1], [\mathfrak{B}, j_1, \tau_1] \models \text{holds}(d)$ , also  $j_1(d) = R(v_1, \dots, v_m)$  und  $\mathfrak{B}, \tau_1 \models j_1(d)$  (Punkte 3 und 4 von Definition 3.2.9). Folglich gilt  $\mathfrak{B}, \tau_1 \models R(v_1, \dots, v_m)$ . Aus der Definition von  $\mathfrak{B}$  folgt dann, dass es  $j_2, \tau_2$  gibt, so dass  $[\mathfrak{A}, j_{(\tau_1(v_1), \dots, \tau_1(v_m))}, \tau_{(\tau_1(v_1), \dots, \tau_1(v_m))}], [\mathfrak{A}, j_2, \tau_2] \models D$ . Per definitionem gilt  $j_{(\tau_1(v_1), \dots, \tau_1(v_m))} = j_1$  und  $\tau_{(\tau_1(v_1), \dots, \tau_1(v_m))} = \tau_1$ , so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$

„ $\Leftarrow$ “: Man nehme an, dass es  $j_2, \tau_2$  gibt, so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$ . Dann gilt  $[\mathfrak{A}, j_{(\tau_1(v_1), \dots, \tau_1(v_m))}, \tau_{(\tau_1(v_1), \dots, \tau_1(v_m))}], [\mathfrak{A}, j_2, \tau_2] \models D$ , woraus aufgrund der Definition von  $\mathfrak{B}$  folgt, dass  $\mathfrak{B}, \tau_1 \models R(v_1, \dots, v_m)$ . Zusammen mit Annahme c) lässt sich daraus folgern, dass  $[\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$ .

$\Theta = \langle C := D \rangle + \Sigma$  und  $C := D$  definiert  $R(v_1, \dots, v_m)$  mittels  $d_1, \dots, d_m, d$ :  $\mathfrak{B}$  wird wie im vorherigen Fall definiert. Wir beweisen die beiden Richtungen des zu beweisenden Bikonditional getrennt:

„ $\Rightarrow$ “:

$$\mathfrak{A}', \sigma \models FF(\langle C := D \rangle + \Sigma, \Delta, l, k)$$

$\Rightarrow \mathfrak{B}', \sigma \models FF(\Sigma, \Delta', l, k)$  für  $\Delta' := \Delta \cup \{\langle R, \lambda v_m \dots \lambda v_1. FF(\langle D \rangle, \Delta, k', k') \rangle\}$ , wobei  $k'$  so gewählt ist, dass  $\Sigma \neq \langle \rangle$ ,  $FF(\Sigma_C, \Delta, k, k')$  definiert ist.

$\Rightarrow$  es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_n, j_1, \dots, j_m, \tau_1, \dots, \tau_m$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q)$  für  $1 \leq q \leq n$  (wobei  $\mathfrak{A}_0 := \mathfrak{B}, \tau_0 := \sigma, j_0 := k_\sigma \cup (\iota \circ \delta)$ ), wegen Punkt 2 der Induktionsannahme: Für diese Anwendung der Induktionsannahme muss allerdings gezeigt werden, dass die folgende Bedingung gilt:

$$\mathfrak{B}', \sigma \cup \iota \models \forall v_1, \dots, v_{|R'|} (R'(v_1, \dots, v_{|R'|}) \leftrightarrow \Delta'(R')(v_1) \dots (v_{|R'|})) \text{ für alle } R' \in Def(\Delta')$$

Dabei gilt die Bedingung wegen Annahme 5.2 für alle  $R' \in Def(\Delta)$ . Es muss also nur noch gezeigt werden, dass die Aussage auch für  $R$  gilt:

$$\mathfrak{B}', \sigma \cup \iota \models \forall v_1, \dots, v_m (R(v_1, \dots, v_m) \leftrightarrow FF(\langle D \rangle, \Delta, k', k'))$$

**Beweis:** Da  $\sigma \cup \iota$  nicht auf  $v_1, \dots, v_m$  definiert ist, ist zu zeigen, dass für jede Erweiterung  $\tau$  von  $\sigma \cup \iota$  auf  $v_1, \dots, v_m$ ,  $\mathfrak{B}', \tau \models R(v_1, \dots, v_m)$  genau dann gilt, wenn  $\mathfrak{B}', \tau \models FF(\langle D \rangle, \Delta, k', k')$  gilt. Sei  $\tau$  eine Erweiterung von  $\sigma \cup \iota$  auf  $v_1, \dots, v_m$ .

$$\mathfrak{B}', \tau \models R(v_1, \dots, v_m)$$

$\Leftrightarrow$  es gibt  $j_2, \tau_2$ , so dass  $[\mathfrak{A}, j_{(\tau(v_1), \dots, \tau(v_m))}, \tau_{(\tau(v_1), \dots, \tau(v_m))}], [\mathfrak{A}, j_2, \tau_2] \models D$  (wegen der Definition von  $\mathfrak{B}$ )

$\Leftrightarrow \mathfrak{A}', \tau_{(\tau(v_1), \dots, \tau(v_m))} \cup \iota \models FF(\langle D \rangle, \Delta, k', k')$  wegen Punkt 2 der Induktionsannahme für „ $\Theta$ “ =  $\langle D \rangle$  (da  $j_{(\tau(v_1), \dots, \tau(v_m))} = k'_{\tau_{(\tau(v_1), \dots, \tau(v_m))}} \cup (\iota \circ \delta)$ , wie leicht gezeigt werden kann)

$\Leftrightarrow \mathfrak{B}', \tau \models FF(\langle D \rangle, \Delta, k', k')$ , da  $FF(\langle D \rangle, \Delta, k', k')$  nicht das Zeichen  $R$  enthält, und da  $\tau = \tau_{(\tau(v_1), \dots, \tau(v_m))} \cup \iota$ .  $\square$

$\Rightarrow$  es gibt  $\mathfrak{A}_0, \dots, \mathfrak{A}_n, j_0, \dots, j_n, \tau_0, \dots, \tau_n$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q+1)$  für  $0 \leq q \leq n$  (wobei  $\mathfrak{A}_{-1} = \mathfrak{A}, \tau_{-1} := \sigma, j_{-1} := k_\sigma \cup (\iota \circ \delta)$ ), aufgrund der oben bewiesenen Tatsache, dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma] \models C := D$ .

„ $\Leftarrow$ “:

Es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_{n+1}, j_1, \dots, j_{n+1}, \tau_1, \dots, \tau_{n+1}$ , so dass

$$[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q) \text{ für } 1 \leq q \leq n+1 \quad (5.4)$$

(wobei  $\mathfrak{A}_0 = \mathfrak{A}, \tau_0 := \sigma, j_0 := k_\sigma \cup (\iota \circ \delta)$ ).

$\Rightarrow \mathfrak{A}'_1, \sigma \cup \iota \models FF(\Sigma, \Delta', l, k)$  wegen Punkt 2 der Induktionsannahme für „ $\Theta$ “ =  $\Sigma$  (denn 5.4 nimmt für  $q = 1$  die Form

$$[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}_0, k_\sigma \cup (\iota \circ \delta), \sigma] \models C := D \quad (5.5)$$

an, da der zweite und dritte Wert des Kontext-Tripels sich bei einer Definitionsbedingung nicht ändert; daher gilt  $\tau_1 := \sigma$  und  $j_1 := k_\sigma \cup (\iota \circ \delta)$ ).

$\Rightarrow \mathfrak{A}', \sigma \cup \iota \models FF(\Sigma, \Delta', l, k)$ , denn  $\mathfrak{A}'_1$  und  $\mathfrak{A}$  unterscheiden sich wegen 5.5 nur beim Zeichen  $R$ , welches nicht in  $FF(\Sigma, \Delta', l, k)$  vorkommt.

$\Rightarrow \mathfrak{A}', \sigma \cup \iota \models FF(\langle C := D \rangle + \Sigma, \Delta, l, k)$  per definitionem des Formelbildes

$\Theta = \langle C := D \rangle$  und  $C := D$  definiert  $p(d_1, \dots, d_m)$  mit Variablen  $v_{i_1}, \dots, v_{i_n}$ :  
 $\mathfrak{A}', \sigma \cup \iota \models FF(\langle C := D \rangle, \Delta, l, k)$  gilt in jedem Falle. Außerdem gilt

$$[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma] \models C := D$$

wobei  $\mathfrak{B}$  als

$$\mathfrak{A} \bowtie \langle p, \{(x_1, \dots, x_m) \in |\mathfrak{A}|^m : \text{es gibt } j_2, \tau_2, \text{ so dass } [\mathfrak{A}, j_{(x_1, \dots, x_m)}, \tau_{(x_{i_1}, \dots, x_{i_n})}], [\mathfrak{A}, j_2, \tau_2] \models D\} \rangle$$

definiert wird,  $j_{(x_1, \dots, x_m)}$  eine Erweiterung von  $k_\sigma \cup (\iota \circ \delta)$  auf die Diskursreferenten von  $C$  ist, so dass  $j_{(x_1, \dots, x_m)}(d_i) := x_i$  für  $1 \leq i \leq m$ , und  $\tau_{(x_1, \dots, x_m)}$  eine Erweiterung von  $\sigma$  auf  $v_{i_1}, \dots, v_{i_n}$  ist, so dass  $\tau_{(x_1, \dots, x_m)}(v_{i_r}) := x_{i_r}$  für alle  $1 \leq r \leq n$ .

### Beweis:

Seien  $j_1$  und  $\tau_1$  Funktionen, für die

- $(k_\sigma \cup (\iota \circ \delta))[d_1, \dots, d_m]j_1$
- $\sigma[v_{i_1}, \dots, v_{i_n}]\tau_1$
- $[\mathfrak{B}, j_1, \tau_1], [\mathfrak{B}, j_1, \tau_1] \models c'$  für jede *math-id*-Bedingung  $c'$  in  $C$

gilt. Zu zeigen ist wieder folgendes:  $[\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$  genau dann, wenn es  $j_2, \tau_2$  gibt, so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$ .

„ $\Rightarrow$ “: Angenommen  $[\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$ . Dann gilt  $[\mathfrak{B}, j_1, \tau_1], [\mathfrak{B}, j_1, \tau_1] \models p(d_1, \dots, d_m)$ . Aus der Definition von  $\mathfrak{B}$  folgt dann, dass es  $j_2, \tau_2$  gibt, so dass

$$[\mathfrak{A}, j_{(j_1(d_1), \dots, j_1(d_m))}, \tau_{(j_1(d_{i_1}), \dots, j_1(d_{i_n}))}], [\mathfrak{A}, j_2, \tau_2] \models D.$$



Wegen Bedingung c) ist  $j_1(d_{i_r}) = \tau_1(v_{i_r})$  für  $1 \leq r \leq n$ , also

$$[\mathfrak{A}, j_{(j_1(d_1), \dots, j_1(d_m))}, \tau_{(\tau_1(v_{i_1}), \dots, \tau_1(v_{i_n}))}], [\mathfrak{A}, j_2, \tau_2] \models D.$$

Per definitionem gilt  $j_{(j_1(d_1), \dots, j_1(d_m))} = j_1$  und  $\tau_{(\tau_1(v_{i_1}), \dots, \tau_1(v_{i_n}))} = \tau_1$ , so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$ .

„ $\Leftarrow$ “: Man nehme an, dass es  $j_2, \tau_2$  gibt, so dass  $[\mathfrak{A}, j_1, \tau_1], [\mathfrak{A}, j_2, \tau_2] \models D$ . Dann gilt  $[\mathfrak{A}, j_{(j_1(d_1), \dots, j_1(d_m))}, \tau_{(\tau_1(v_{i_1}), \dots, \tau_1(v_{i_n}))}], [\mathfrak{A}, j_2, \tau_2] \models D$ , woraus aufgrund der Definition von  $\mathfrak{B}$  folgt, dass  $\mathfrak{B}, \tau_1 \models p(d_1, \dots, d_m)$ . Zusammen mit Annahme c) lässt sich daraus folgern, dass  $[\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, j_1, \tau_1] \models C$ .

$\Theta = \langle C := D \rangle + \Sigma$  und  $C := D$  definiert  $p(d_1, \dots, d_m)$  mittels  $\alpha_1, \dots, \alpha_n$  und mit Variablen  $v_{i_1}, \dots, v_{i_n}$ :

$\mathfrak{B}$  wird wie im vorherigen Fall definiert.

„ $\Rightarrow$ “:

$$\mathfrak{A}', \sigma \models FF(\langle C := D \rangle + \Sigma, \Delta, l, k)$$

$$\Rightarrow \mathfrak{B}', \sigma \models FF(\Sigma, \Delta', l, k) \text{ für}$$

$$\Delta' := \Delta \cup \{ \langle p, \lambda\delta(\alpha_m) \dots \lambda\delta(\alpha_1).FF(\langle D \rangle, \Delta, k', k') \rangle \},$$

wobei  $k'$  so gewählt ist, dass  $\Sigma \neq \langle \rangle$ ,  $FF(\Sigma_C, \Delta, k, k')$  definiert ist.

$\Rightarrow$  es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_n, j_1, \dots, j_n, \tau_1, \dots, \tau_n$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q)$  für  $1 \leq q \leq n$  (wobei  $\mathfrak{A}_0 := \mathfrak{B}, \tau_0 := \sigma, j_0 := k_\sigma \cup (\iota \circ \delta)$ ), wegen Punkt 2 der Induktionsannahme: Für diese Anwendung der Induktionsannahme muss gezeigt werden, dass die folgende Bedingung gilt:

$$\mathfrak{B}', \sigma \cup \iota \models \forall x_1, \dots, x_{|R|} (R(x_1, \dots, x_{|R|}) \leftrightarrow \Delta'(R)(x_1) \dots (x_{|R|})) \text{ für alle } R \in Def(\Delta')$$

Dabei gilt die Bedingung wegen Annahme 5.2 für alle  $R \in Def(\Delta)$ . Es muss also nur noch gezeigt werden, dass die Aussage auch für  $p$  gilt:

$$\mathfrak{B}', \sigma \cup \iota \models \forall \delta(\alpha_1), \dots, \delta(\alpha_m) (p(\delta(\alpha_1), \dots, \delta(\alpha_m)) \leftrightarrow FF(\langle D \rangle, \Delta, k', k'))$$

**Beweis:** Da  $\sigma \cup \iota$  nicht auf  $\delta(\alpha_1), \dots, \delta(\alpha_m)$  definiert ist (wie leicht gezeigt werden kann), ist zu zeigen, dass für jede Erweiterung  $\tau$  von  $\sigma \cup \iota$  auf  $\delta(\alpha_1), \dots, \delta(\alpha_m)$ ,  $\mathfrak{B}', \tau \models p(\delta(\alpha_1), \dots, \delta(\alpha_m))$  genau dann gilt, wenn  $\mathfrak{B}', \tau \models FF(\langle D \rangle, \Delta, k', k')$  gilt. Sei  $\tau$  eine Erweiterung von  $\sigma \cup \iota$  auf  $\delta(\alpha_1), \dots, \delta(\alpha_m)$ . Man beachte, dass jedes  $\delta(\alpha_i)$  entweder  $\delta_{d_i}$  oder  $v_i$  sein kann. Im Beweis muss unterschieden werden zwischen dem Teil von  $\tau$ , der schon in  $\sigma \cup \iota$  ist, jenem Teil  $\tau_0$  von  $\tau$ , der der Erweiterung auf Variablen der Form  $\delta_{d_i}$  entspricht, und jenem Teil  $\tau_1$  von  $\tau$ , der der Erweiterung auf Variablen von  $L$  entspricht:

- $\tau_0 := \tau|_{\{\delta(\alpha_i) : \delta(\alpha_i) = \delta_{d_i}\}}$
- $\tau_1 := \tau|_{\{v_{i_1}, \dots, v_{i_r}\}}$

Es gilt also  $\tau = \sigma \cup \iota \cup \tau_0 \cup \tau_1$ .

$\mathfrak{B}', \tau \models p(\delta(\alpha_1), \dots, \delta(\alpha_m))$

$\Leftrightarrow$  es gibt  $j_2, \tau_2$ , so dass  $[\mathfrak{A}, j(\tau(\delta(\alpha_1)), \dots, \tau(\delta(\alpha_m))), \tau(\tau(v_{i_1}), \dots, \tau(v_{i_n}))]$ ,  $[\mathfrak{A}, j_2, \tau_2] \models D$  (wegen der Definition von  $\mathfrak{B}$ )

$\Leftrightarrow \mathfrak{A}', (\sigma \cup \tau_1) \cup (\iota \cup \tau_0) \models FF(\langle D \rangle, \Delta, k', k')$  wegen Punkt 2 der Induktionsannahme für „ $\Theta$ “ =  $\langle D \rangle$  (denn  $j(\tau(\delta_{d_1}), \dots, \tau(\delta_{d_m})) = k'_\sigma \cup ((\iota \cup \tau_0) \circ \delta)$  und  $\tau(\tau(v_{i_1}), \dots, \tau(v_{i_n})) = (\sigma \cup \tau_1)$ , wie leicht gezeigt werden kann)

$\Leftrightarrow \mathfrak{B}', \tau \models FF(\langle D \rangle, \Delta, k', k')$ , da  $FF(\langle D \rangle, \Delta, k', k')$  nicht das Zeichen  $p$  enthält.  $\square$

$\Rightarrow$  es gibt  $\mathfrak{A}_0, \dots, \mathfrak{A}_n, j_0, \dots, j_n, \tau_0, \dots, \tau_n$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q)$  für  $0 \leq q \leq n$  (wobei  $\mathfrak{A}_{-1} = \mathfrak{A}, \tau_{-1} := \sigma, j_{-1} := k_\sigma \cup (\iota \circ \delta)$ ), aufgrund der oben bewiesenen Tatsache, dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{B}, k_\sigma \cup (\iota \circ \delta), \sigma] \models C := D$ .

„ $\Leftarrow$ “:

Es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_{n+1}, j_1, \dots, j_{n+1}, \tau_1, \dots, \tau_{n+1}$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q)$  für  $1 \leq q \leq n+1$  (wobei  $\mathfrak{A}_0 = \mathfrak{A}, \tau_0 := \sigma, j_0 := k_\sigma \cup (\iota \circ \delta)$ ).

$\Rightarrow \mathfrak{A}'_1, \sigma \cup \iota \models FF(\Sigma, \Delta', l, k)$  wegen Punkt 2 der Induktionsannahme für „ $\Theta$ “ =  $\Sigma$   
 $\Rightarrow \mathfrak{A}', \sigma \cup \iota \models FF(\Sigma, \Delta', l, k)$ , denn  $\mathfrak{A}'_1$  und  $\mathfrak{A}$  unterscheiden sich nur beim Zeichen  $p$ , welches nicht in  $FF(\Sigma, \Delta', l, k)$  vorkommt.

$\Rightarrow \mathfrak{A}', \sigma \cup \iota \models FF(\langle C := D \rangle + \Sigma, \Delta, l, k)$

$\Theta = \langle B \rangle$  für eine PRS  $B$

$\mathfrak{A}', \sigma \cup \iota \models FF(\langle B \rangle, \Delta, k, k)$

$\Leftrightarrow$  es gibt ein  $l'$ , so dass  $\mathfrak{A}', \sigma \cup \iota \models \exists X_{B, l'} FF(\Sigma_B, \Delta, k, l')$

$\Leftrightarrow$  es gibt ein  $l'$  und ein  $\tau \supseteq \sigma \cup \iota$  mit  $Def(\tau) = Def(\sigma \cup \iota) \cup X_{B, l'}$ , so dass  $\mathfrak{A}', \tau \models FF(\Sigma_B, \Delta, k, l')$ . Für diesen Schritt müssen wir zeigen, dass  $Def(\sigma \cup \iota) \cap X_{B, l'} = \emptyset$ :

- $Def(\sigma) = Def(k)$  und  $Z_B = Def(l') \setminus Def(k)$ , so dass  $Def(\sigma) \cap Z_B = \emptyset$ .
- $Def(\iota) \cap \delta[D(\theta)] = \emptyset$  und  $\delta[D(\theta)] \subseteq Y_{B, l'}$ , woraus folgt, dass  $Def(\iota) \cap Y_{B, l'} = \emptyset$ .

$\Leftrightarrow$  es gibt ein  $l'$ , so dass es ein  $\eta : Y_{B, l'} \rightarrow |\mathfrak{A}|$  und ein  $\tau' \supseteq \sigma$  gibt, für welche  $Def(\tau') = Def(\sigma) \cup Z_B$  und  $\mathfrak{A}', \tau' \cup (\iota \cup \eta) \models FF(\Sigma_B, \Delta, k, l')$  gilt (hier wurde das  $\tau$  vom vorherigen Schritt von seinem  $\iota$ -Teil bereinigt und in ein  $\tau'$  und dann ein  $\eta$  aufgeteilt).

$\Leftrightarrow$  es gibt ein  $l'$ , so dass  $FF(\Sigma_B, \Delta, k, l')$  definiert ist, und so dass es ein  $\eta : Y_{B, l'} \rightarrow |\mathfrak{A}|$  und ein  $\tau' \supseteq \sigma$  mit den folgenden Eigenschaften gibt:

- $Def(\tau') = Def(\sigma) \cup Z_B$

- es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_m, j_1, \dots, j_m, \tau_1, \dots, \tau_m$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma_B(q)$  für  $1 \leq q \leq m$  (wobei  $m := \text{Länge}(\Sigma_B)$  und  $\mathfrak{A}_0 = \mathfrak{A}, \tau_0 := \tau', j_0 := l'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ ); hierbei wurde Punkt 2 der Induktionsannahme mit „ $\Theta$ “ =  $\Sigma_B$  angewendet. Für diese Anwendung der Induktionsannahme müssen wir zeigen, dass

1.  $\xi(\Sigma_B, l', \Delta) \subseteq \text{Def}(\iota \cup \eta) \subseteq \{\delta_n : n \in \mathbb{N}\}$ ,
2.  $\text{Def}((\iota \cup \eta) \circ \delta) \cap (\text{Def}(l') \cup D(\Sigma_B)) = \emptyset$ ,
3.  $\text{Bild}(\iota \cup \eta) \subseteq |\mathfrak{A}|$ .

Dies lässt sich auf ähnliche Weise zeigen wie die analoge Aussage im Fall  $\Theta = B \Rightarrow C$ .

$\Leftrightarrow$  es gibt ein  $l'$ , so dass  $FF(\Sigma_B, \Delta, k, l')$  definiert ist, und so dass es ein  $\eta : Y_{B, l'} \rightarrow |\mathfrak{A}|$  und ein  $\tau' \supseteq \sigma$  gibt, für welche  $\text{Def}(\tau') = \text{Def}(\sigma) \cup Z_B$  ist und  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt (wobei  $j_0 := l'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ ); für diesen Schritt wurde Punkt 1 von Definition 3.2.9 angewendet: es kann leicht gezeigt werden, dass  $k_\sigma \cup (\iota \circ \delta)[d_1, \dots, d_{n_1}]j_0$  und  $\sigma[v_1, \dots, v_{n_2}]\tau'$ , wobei  $d_1, \dots, d_{n_1}$  die Diskursreferenten von  $B$  sind, und  $v_1, \dots, v_{n_2}$  die Variablenreferenten von  $B$  sind).

$\Leftrightarrow$  es gibt  $j_0$  und  $\tau'$ , so dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt

Für die „ $\Leftarrow$ “-Richtung dieses letzten Bikonditionals beachte man, dass es immer ein  $l'$  gibt, so dass  $FF(\Sigma_B, \Delta, k, l')$  definiert ist, und dass für alle  $\tau' \supseteq \sigma, j_0 \supseteq k_\sigma \cup (\iota \circ \delta)$ , so dass  $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt,  $\text{Def}(\tau') = \text{Def}(\sigma) \cup Z_B$  ist und es ein  $\eta : Y_{B, l'} \rightarrow |\mathfrak{A}|$  gibt, so dass  $j_0$  die Form  $k'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$  hat (vergleiche den letzten Bikonditional im Falle  $\Theta = B \Rightarrow C$ ).

$\Theta = \langle B \rangle + \Sigma$  für eine PRS  $B$

$\mathfrak{A}', \sigma \cup \iota \models FF(\langle B \rangle + \Sigma, \Delta, l, k)$

$\Leftrightarrow$  es gibt ein  $k'$ , so dass  $l \cap (k' \setminus k) = \emptyset$  und  $\mathfrak{A}', \sigma \cup \iota \models \exists X_{B, k'}(FF(\Sigma_B, \Delta, k, k') \wedge FF(\Sigma, \Delta, l \cup (k' \setminus k), k'))$ .

$\Leftrightarrow$  es gibt ein  $k'$ , so dass  $l \cap (k' \setminus k) = \emptyset$  ist, und ein  $\tau \supseteq \sigma \cup \iota$  mit  $\text{Def}(\tau) = \text{Def}(\sigma \cup \iota) \cup X_{B, k'}$ , so dass  $\mathfrak{A}', \tau \models FF(\Sigma_B, \Delta, k, k')$  und  $\mathfrak{A}', \tau \models FF(\Sigma, \Delta, l \cup (k' \setminus k), k')$ .

Für diesen Schritt müssen wir zeigen, dass  $\text{Def}(\sigma \cup \iota) \cap X_{B, k'} = \emptyset$  :

- $\text{Def}(\sigma) = \text{Def}(k)$  und  $Z_B = \text{Def}(k') \setminus \text{Def}(k)$ , so dass  $\text{Def}(\sigma) \cap Z_B = \emptyset$ .
- $\text{Def}(\iota) \cap \delta[D(\theta)] = \emptyset$  und  $\delta[D(\theta)] \subseteq Y_{B, k'}$ , woraus folgt, dass  $\text{Def}(\iota) \cap Y_{B, k'} = \emptyset$ .

$\Leftrightarrow$  es gibt ein  $k'$ , so dass  $l \cap (k' \setminus k) = \emptyset$  ist, und so dass es ein  $\eta : Y_{B,k'} \rightarrow |\mathfrak{A}|$  und ein  $\tau' \supseteq \sigma$  gibt, für welche  $Def(\tau') = Def(\sigma) \cup Z_B$ ,  $\mathfrak{A}', \tau' \cup (\iota \cup \eta) \models FF(\Sigma_B, \Delta, k, k')$  und  $\mathfrak{A}', \tau' \cup (\iota \cup \eta) \models FF(\Sigma, \Delta, l \cup (k' \setminus k), k')$  gilt (hier wurde das  $\tau$  vom vorherigen Schritt von seinem  $\iota$ -Teil bereinigt und in ein  $\tau'$  und dann ein  $\eta$  aufgeteilt).

$\Leftrightarrow$  es gibt ein  $k'$ , so dass  $l \cap (k' \setminus k) = \emptyset$  ist, so dass  $FF(\Sigma_B, \Delta, k, k')$  und  $FF(\Sigma, \Delta, l \cup (k' \setminus k), k')$  definiert sind, und so dass es ein  $\eta : Y_{B,k'} \rightarrow |\mathfrak{A}|$  und ein  $\tau' \supseteq \sigma$  mit den folgenden Eigenschaften gibt:

- $Def(\tau') = Def(\sigma) \cup Z_B$
- es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_m, j_1, \dots, j_m, \tau_1, \dots, \tau_m$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma_B(q)$  für  $1 \leq q \leq m$  (wobei  $m := \text{Länge}(\Sigma_B)$  und  $\mathfrak{A}_0 = \mathfrak{A}, \tau_0 := \tau', j_0 := k'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ ; hierbei wurde Punkt 2 der Induktionsannahme mit „ $\Theta$ “ =  $\Sigma_B$  angewendet; für diese Anwendung der Induktionsannahme müssen wir zeigen, dass
  1.  $\xi(\Sigma_B, k', \Delta) \subseteq Def(\iota \cup \eta) \subseteq \{\delta_n : n \in \mathbb{N}\}$ ,
  2.  $Def((\iota \cup \eta) \circ \delta) \cap (Def(k') \cup D(\Sigma_B)) = \emptyset$ ,
  3.  $Bild(\iota \cup \eta) \subseteq |\mathfrak{A}|$ ;

dies lässt sich auf ähnliche Weise zeigen wie die analoge Aussage im Fall  $\Theta = B \Rightarrow C$ .

- es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_n, j_1, \dots, j_n, \tau_1, \dots, \tau_n$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q)$  für  $1 \leq q \leq n$  (wobei  $\mathfrak{A}_0 = \mathfrak{A}, \tau_0 := \tau', j_0 := k'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ ; hierbei wurde Punkt 2 der Induktionsannahme mit „ $\Theta$ “ =  $\Sigma$  angewendet; für diese Anwendung der Induktionsannahme müssen wir zeigen, dass
  1.  $\xi(\Sigma, k', \Delta) \subseteq Def(\iota \cup \eta) \subseteq \{\delta_n : n \in \mathbb{N}\}$ ,
  2.  $Def((\iota \cup \eta) \circ \delta) \cap (Def(k') \cup D(\Sigma)) = \emptyset$ ,
  3.  $Bild(\iota \cup \eta) \subseteq |\mathfrak{A}|$ ;

dies lässt sich auf ähnliche Weise zeigen wie die analoge Aussage im Fall  $\Theta = B \Rightarrow C$ .

$\Leftrightarrow$  es gibt ein  $k'$ , so dass  $l \cap (k' \setminus k) = \emptyset$  ist, so dass  $FF(\Sigma_B, \Delta, k, k')$  und  $FF(\Sigma, \Delta, l \cup (k' \setminus k), k')$  definiert sind, und so dass es ein  $\eta : Y_{B,k'} \rightarrow |\mathfrak{A}|$  und ein  $\tau' \supseteq \sigma$  mit den folgenden Eigenschaften gibt:

- $Def(\tau') = Def(\sigma) \cup Z_B$
- $[\mathfrak{A}, k_\sigma \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  (wobei  $j_0 := k'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ ; für diesen Schritt wurde Punkt 1 von Definition 3.2.9 angewendet: es kann leicht gezeigt werden, dass  $k_\sigma \cup (\iota \circ \delta)[d_1, \dots, d_n]j_0$  und  $\sigma[v_1, \dots, v_n]\tau'$ , wobei  $d_1, \dots, d_n$  die Diskursreferenten von  $B$  sind, und  $v_1, \dots, v_n$  die Variablenreferenten von  $B$  sind)

- es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_n, j_1, \dots, j_n, \tau_1, \dots, \tau_n$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q)$  für  $1 \leq q \leq n$  (wobei  $\mathfrak{A}_0 = \mathfrak{A}, \tau_0 := \tau', j_0 := k'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$ )

$\Leftrightarrow$  es gibt  $\mathfrak{A}_0, \dots, \mathfrak{A}_n, j_0, \dots, j_n, \tau_0, \dots, \tau_n$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q+1)$  für  $0 \leq q \leq n$  (wobei  $\mathfrak{A}_{-1} = \mathfrak{A}, \tau_{-1} := \sigma, j_{-1} := k_{\sigma} \cup (\iota \circ \delta)$ )

Für die „ $\Leftarrow$ “-Richtung dieses letzten Bikonditionals beachte man, dass weil  $FF(\langle B \rangle + \Sigma, \Delta, l, k)$  definiert ist, es im jeden Falle ein  $k'$  gibt, so dass  $l \cap (k' \setminus k) = \emptyset$  ist, und so dass  $FF(\Sigma_B, \Delta, k, k')$  und  $FF(\Sigma, \Delta, l \cup (k' \setminus k), k')$  definiert sind. Außerdem beachte man, dass für alle  $\tau' > \sigma, j_0 > k_{\sigma} \cup (\iota \circ \delta)$ , so dass  $[\mathfrak{A}, k_{\sigma} \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, j_0, \tau'] \models B$  gilt,  $Def(\tau') = Def(\sigma) \cup Z_B$  ist und es ein  $\eta : Y_{B,l} \rightarrow |\mathfrak{A}|$  gibt, so dass  $j_0$  die Form  $k'_{\tau'} \cup ((\iota \cup \eta) \circ \delta)$  hat (vergleiche den letzten Bikonditional im Falle  $\Theta = B \Rightarrow C$ ).

$\Theta = \langle c \rangle$  für eine PRS-Bedingung  $c$ , die nicht PRS, Definitionsbedingung oder *math\_id*-Bedingung ist

$$\mathfrak{A}', \sigma \cup \iota \models FF(\langle c \rangle, \Delta, l, k)$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models FB(c, \Delta, k)$$

$$\Leftrightarrow [\mathfrak{A}, k_{\sigma} \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_{\sigma} \cup (\iota \circ \delta), \sigma] \models c \text{ wegen Punkt 1 der Induktionsannahme}$$

$\Leftrightarrow$  es gibt  $\mathfrak{A}_1, j_1, \tau_1$ , so dass  $[\mathfrak{A}, k_{\sigma} \cup (\iota \circ \delta), \sigma], [\mathfrak{A}_1, j_1, \tau_1] \models c$ , da  $c$  keine Definitionsbedingung ist, und somit  $c$  von keiner Veränderung der Struktur im Kontexttripel erfüllt werden kann.

$\Theta = \langle c \rangle + \Sigma$  für eine PRS-Bedingung  $c$ , die nicht PRS, Definitionsbedingung oder *math\_id*-Bedingung ist

$$\mathfrak{A}', \sigma \cup \iota \models FF(\langle c \rangle + \Sigma, \Delta, l, k)$$

$$\Leftrightarrow \mathfrak{A}', \sigma \cup \iota \models (FB(c, \Delta, k) \wedge FF(\Sigma, \Delta, l, k))$$

$\Leftrightarrow [\mathfrak{A}, k_{\sigma} \cup (\iota \circ \delta), \sigma], [\mathfrak{A}, k_{\sigma} \cup (\iota \circ \delta), \sigma] \models c$  (Punkt 1 der Induktionsannahme), und es gibt  $\mathfrak{A}_2, \dots, \mathfrak{A}_{n+1}, j_2, \dots, j_{n+1}, \tau_2, \dots, \tau_{n+1}$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Sigma(q-1)$  für  $2 \leq q \leq n+1$ , wobei  $\mathfrak{A}_1 := \mathfrak{A}, \tau_1 := \sigma, j_1 := k_{\sigma} \cup (\iota \circ \delta)$  (Punkt 2 der Induktionsannahme)

$\Leftrightarrow$  es gibt  $\mathfrak{A}_1, \dots, \mathfrak{A}_{n+1}, j_1, \dots, j_{n+1}, \tau_1, \dots, \tau_{n+1}$ , so dass  $[\mathfrak{A}_{q-1}, j_{q-1}, \tau_{q-1}], [\mathfrak{A}_q, j_q, \tau_q] \models \Theta(q)$  für  $1 \leq q \leq n+1$ , wobei  $\mathfrak{A}_0 := \mathfrak{A}, \tau_0 := \sigma, j_0 := k_{\sigma} \cup (\iota \circ \delta)$  („ $\Rightarrow$ “ trivial; „ $\Leftarrow$ “ gilt, weil  $c$  eine PRS-Bedingung ist, die weder Definitionsbedingung noch PRS ist, und somit  $c$  nicht den Kontext ändert, also  $[\mathfrak{A}_0, j_0, \tau_0] = [\mathfrak{A}_1, j_1, \tau_1]$  ist)  $\square$

**Theorem 5.2.2.** Sei  $B$  eine wohlgeformte PRS. Sei  $\mathfrak{A}$  eine  $(L, \Lambda)$ -Struktur.  $\mathfrak{A}$  verifiziert  $B$  genau dann, wenn  $\mathfrak{A}' \models FP(B)$ .

**Beweis**

$\mathfrak{A}$  verifiziert  $B$

$\Leftrightarrow$  es gibt eine  $|\mathfrak{A}|$ -Teilbelegung  $\sigma$  und eine Einbettung  $i: \mathbb{N} \rightsquigarrow F_L \cup U$ , so dass  $[\mathfrak{A}, \emptyset, \emptyset], [\mathfrak{A}, i, \sigma] \models B$  gilt

$\Leftrightarrow \mathfrak{A}', \emptyset \models FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)$  wegen Punkt 2 von Lemma 5.2.1. Für diese Anwendung des Lemmas müssen wir noch zeigen, dass die Bedingungen für das Lemma erfüllt sind:

1.  $Def(\emptyset) \cap D(\langle B \rangle) = \emptyset$  ist trivial.
2.  $FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)$  ist definiert, da  $FP(B)$  definiert ist (was wir durch Theorem 4.4.4 wissen).
3.  $\xi(\langle B \rangle, \emptyset, \emptyset) \subseteq Def(\emptyset) \subset \{\delta_n : n \in \mathbb{N}\}$ , denn nach Theorem 4.4.4 ist  $frei(FP(B)) = frei(FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)) = \emptyset$ , so dass  $\xi(\langle B \rangle, \emptyset, \emptyset) = \emptyset$  ist.
4.  $Def(\emptyset \circ \delta) \cap (Def(\emptyset) \cup D(\xi(\langle B \rangle, \emptyset, \emptyset))) = \emptyset$  ist trivial.
5.  $Bild(\emptyset) \subseteq |\mathfrak{A}|$  ist trivial.
6. Trivialerweise gilt für alle  $R \in Def(\emptyset)$

$$\mathfrak{A}', \emptyset \models \forall x_1, \dots, x_{|R|} (R(x_1, \dots, x_{|R|}) \leftrightarrow \emptyset(R)(x_1) \dots (x_{|R|})).$$

$\Leftrightarrow \mathfrak{A}' \models FP(B)$ , da  $FP(B)$  ein  $L$ -Satz ist und gleich  $FF(\langle B \rangle, \emptyset, \emptyset, \emptyset)$  ist.  $\square$

## Kapitel 6

---

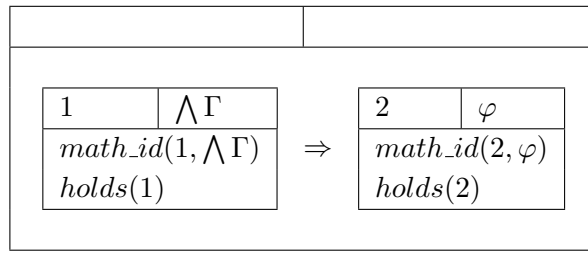
# Ausblick

---

Auch in Zukunft müssen die theoretischen Aspekte von PRSen noch weiter ausgearbeitet werden. Dabei wird das schon von KÜHLWEIN [9] definierte Naproche-Kalkül, das eine formale Version des im Naproche-System implementierten Überprüfungsalgorithmus darstellt, eine wichtige Rolle spielen. Dieser Algorithmus überprüft, ob eine PRS gültig ist. Vereinfacht ausgedrückt wird dabei die PRS Bedingung für Bedingung durchgearbeitet, und bei jeder Bedingung wird mittels eines automatischen Beweisers überprüft, ob diese aus den vorherigen Bedingungen folgt. Dabei kommen Aspekte der PRS-Definition zum Tragen, die in der vorliegenden Arbeit noch keine Rolle gespielt haben: So werden Theorem-PRSen und in sie eingebettete Ziel-PRSen und Beweis-PRSen besonders behandelt: Erst wird die Beweis-PRS überprüft; die Ziel-PRS wird danach aus den im Beweis gemachten Aussagen abgeleitet.

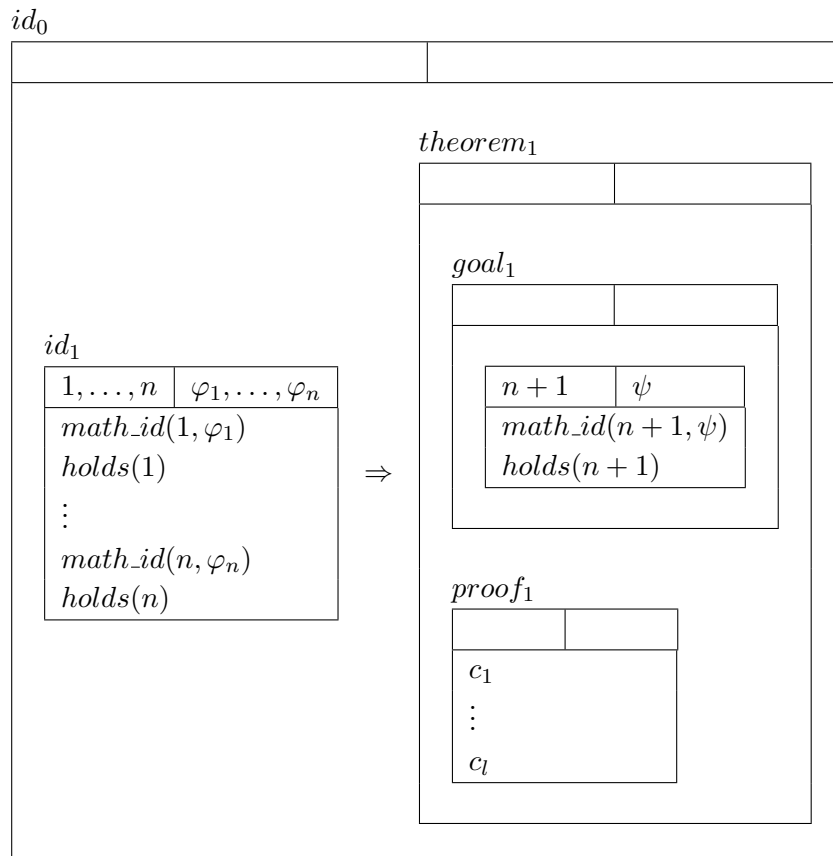
Der automatische Beweiser des Überprüfungsalgorithmus wird im Naproche-Kalkül durch ein (möglicherweise nicht vollständiges) formales Kalkül  $P$  für die Prädikatenlogik erster Stufe modelliert. KÜHLWEIN hat gezeigt, dass aus der Korrektheit von  $P$  die Korrektheit des Naproche-Kalküls folgt. Da es aber noch keine formale PRS-Semantik gab, konnte er die Korrektheit des Naproche-Kalküls nur über sein Formelbild definieren. Mit der in dieser Arbeit vorgestellten PRS-Semantik lässt sich die Korrektheit des Naproche-Kalküls nun modelltheoretisch definieren. Eine Aufgabe für die bevorstehende theoretische Arbeit wäre zu zeigen, dass auch die so definierte Korrektheit des Naproche-Kalküls aus der Korrektheit von  $P$  folgt.

KÜHLWEIN hat die Vollständigkeit des Naproche-Kalküls wie folgt definiert: Das Naproche-Kalkül ist vollständig, wenn für jede logische Schlussfolgerung  $\Gamma \models \varphi$  die folgende PRS vom Naproche-Kalkül akzeptiert wird:



Um so definierte Vollständigkeit zu beweisen, muss man von der Vollständigkeit von  $P$  ausgehen. Das Problem ist, dass  $P$  einen automatischen Beweiser modelliert, und automatische Beweiser (bei festgelegter maximaler Laufzeit) niemals vollständig sein können.

Es ist aber auch möglich, Vollständigkeit des Naproche-Kalküls so zu definieren, dass man auch bei einem unvollständigen  $P$  von einem vollständigen Naproche-Kalkül reden kann: Das Naproche-Kalkül sei vollständig, wenn es für jede logische Schlussfolgerung  $\varphi_1, \dots, \varphi_n \models \psi$  PRS-Bedingungen  $c_1, \dots, c_l$  gibt, so dass die folgende PRS vom Naproche-Kalkül akzeptiert wird:



Diese PRS ist die Übersetzung eines Naproche-Texts der folgenden Form:



Assume that  $\varphi_1$  and ... and  $\varphi_n$ .

Theorem:  $\psi$ .

Proof: ... Qed.

(Wobei der Text zwischen „Proof“ und „Qed“ in die Bedingungen  $c_1, \dots, c_l$  übersetzt wird.)

Damit das Naproche-Kalkül in diesem Sinne vollständig ist, muss  $P$  nur einige einfache logische Schlussfolgerungen beweisen können. Dabei wäre es interessant, minimale Anforderungen an  $P$  zu finden. Diese entsprächen dann minimalen Anforderungen an den automatischen Beweiser des Naproche-Systems, die benötigt werden, damit sich jeder formalisierbare mathematische Beweis als ein überprüfbarer Naproche-Text schreiben lässt, auch wenn mit sehr vielen Beweisschritten.



## Kapitel 7

---

### Fazit

---

In dieser Arbeit wurde erstmals eine formale, modelltheoretische Semantik für Beweisrepräsentationsstrukturen (PRSen) definiert. Diese orientiert sich an der dynamischen Semantikdefinition für Diskursrepräsentationsstrukturen.

Außerdem haben wir eine Übersetzung von PRSen in Prädikatenlogik erster Stufe definiert. Wir haben gezeigt, dass diese als Formelbild bezeichnete Übersetzung wohldefiniert ist, und dass jede wohlgeformte PRS zu ihrem Formelbild äquivalent ist, so dass man das Formelbild als eine korrekte Übersetzung bezeichnen kann. Für den dafür gebrauchten Äquivalenzbegriff war die zuvor definierte PRS-Semantik unabdinglich.

Die in dieser Arbeit behandelten theoretischen Aspekte von PRSen spielen bei der Weiterentwicklung und Implementierung von Naproche eine entscheidende Rolle: Bei der Arbeit an Naproche werden gelegentlich Erweiterungen und Änderungen der PRS-Definition in Betracht gezogen. Dabei ist es wichtig, die theoretischen Implikationen dieser Erweiterungen und Änderungen gut zu verstehen, wozu die vorliegende Arbeit einen wichtigen Beitrag gemacht hat.



---

## Anhang A

---

# Die Grammatik der Naproche-Sprache

---

```
:- module(dcg_simple, []).

:- use_module(library(pldoc)).
:- use_module(naproche(gulp4swi)).
:- use_module(naproche(dcg_simple_lexicon)).
:- use_module(naproche(fo_grammar)).

/** <module> Naproche Grammar
 *
 * This module describes the definite clause grammar for the Naproche language.
 * @author Marcos Cramer
 */

%=====
% Macrostructure
%=====

% BASICS

% A Naproche text consists of axioms, lemmas, theorems and text.

naproche_text --> axiom, naproche_text.
naproche_text --> lemma, naproche_text.
naproche_text --> theorem, naproche_text.
naproche_text --> assumption_naproche_text, naproche_text.
naproche_text --> text, naproche_text.
naproche_text --> [].

% Text contains statements, definitions, assumptions, assumption closings, cases and case closings.
% A text can never be empty.

text --> statement, ['.'], text_tail.
text --> definition, ['.'], text_tail.
text --> assumption_text, text_tail.
text --> assumption, ['.'], text_tail.
text --> cases_text, text_tail.
text --> cases.

% text_tail is just like text, but can be empty.
```

```

text_tail --> statement, ['.'], text_tail.
text_tail --> definition, ['.'], text_tail.
text_tail --> assumption_text, text_tail.
text_tail --> assumption, ['.'], text_tail.
text_tail --> cases_text, text_tail.
text_tail --> cases.
text_tail --> [].

% STATEMENTS
%
% A statement has a proposition coordination as its core, can start with a statement trigger,
% and can have a reference.
% Alternatively, it can just be the 'sentence' "trivial".

statement --> proposition_coord(mode~finite).
statement --> statement_trigger, proposition_coord(mode~finite).
statement --> reference, proposition_coord(mode~finite).
statement --> statement_trigger, reference, proposition_coord(mode~finite).
statement --> proposition_coord(mode~finite), reference.
statement --> statement_trigger, proposition_coord(mode~finite), reference.
statement --> [contradiction].
statement --> [trivial].

% REFERENCES
%
% A reference refers to an axiom, lemma, theorem or proof method, (optionally from a different
% Naproche text).

reference --> [by], [axiom], identifier, optional_comma.
% reference --> [by], [axiom], identifier, [in], naproche_text_id, optional_comma.
reference --> [by], [lemma], identifier, optional_comma.
% reference --> [by], [lemma], identifier, [in], naproche_text_id, optional_comma.
reference --> [by], [theorem], identifier, optional_comma.
% reference --> [by], [theorem], identifier, [in], naproche_text_id, optional_comma.
reference --> [by], [induction], optional_comma.

% DEFINITIONS
%
% Definitions can have a number of possible forms. Here is how to define new predicates symbols,
% new adjectives, new nouns and new verbs.

definition --> [define], definiendum, iff, proposition_coord(mode~finite).

% Here is how to explicitly define new function symbols, including 0-ary (i.e. constants).
% fo_free_function(F,_,_) is defined to be any term of the form F(x1, ..., xn), where F is an
% n-ary function_symbol.
% fo_term_without(F,_,_) is defined to be any term not containing the function symbol F.

definition -->
[define], [math(X)], [to], [be], [math(Y)],
{fo_free_function(F,_,_,X,[]), fo_term_without(F,_,_,Y,[])}.

% Here is how to recursively define new function symbols (e.g. in Naproche-Landau without talk
% about sets and functions).
% fo_induction_start and fo_induction_step are of the required form to for a recursive definition
% of FunctionSymbol.

definition -->
[define], [math([FunctionSymbol])], [recursively], [':'], [math(Y)], [math(Z)],
{
fo_induction_start(FunctionSymbol,Number,_,_,_,Y,[]),
fo_induction_step(FunctionSymbol,Number,_,_,_,Z,[])
}.

definiendum --> [math(X)], {fo_free_predicate_symbol(.,_,X,[])}.
definiendum --> [math(X)], [to], [be], adjective, {fo_variable(.,X,[])}.
definiendum --> [math(X)], [to], [be], indefinite_specifier, noun, {fo_variable(.,X,[])}.
definiendum --> [math(X)], [to], verb(transitive~minus..infinitive~plus), {fo_variable(.,X,[])}.

```

---

```

definiendum -->
[math(X)], [to], verb(transitive~plus..infinitive~plus), [math(Y)],
{fo_variable(_,X,[]), fo_variable(_,Y,[]), \+ X=Y}.

% ASSUMPTIONS

% assumption_naproche_text starts with an assumption and ends with an assumption closing.
% In between there is naproche_text.

assumption_naproche_text --> assumption, ['.'], naproche_text_all_ass_closed, closing.

% assumption_text starts with an assumption and ends with an assumption closing.
% In between there is text.

assumption_text --> assumption, ['.'], text_all_ass_closed, closing.

% An assumption starts with an assumption trigger or with "let" or
% "consider". In the last two cases, the sentence has an infinitive verb.
% This property of the sentence is marked by mode~infinitive.

assumption --> ass_trigger, proposition_coord(mode~finite).
assumption --> [let], proposition_coord(mode~infinitive).
assumption --> [consider], proposition_coord(mode~infinitive).

% Alternatively, an assumption can be used to introduce new variables (possibly fixing some property
% for them, e.g. using a such-that-clause).

assumption --> [consider], variable_list_bar.
assumption --> [fix], variable_list_bar.
assumption --> [let], variable_list, [be], [given].
assumption --> [let], variable_list, [be], [a], nbar.
assumption --> [let], variable_list, [be], [an], nbar.

% text_all_ass_closed is just like text, but without unclosed assumptions.

text_all_ass_closed --> statement, ['.'], text_all_ass_closed_tail.
text_all_ass_closed --> definition, ['.'], text_all_ass_closed_tail.
text_all_ass_closed --> assumption_text, text_all_ass_closed_tail.
text_all_ass_closed --> cases_text, text_all_ass_closed_tail.
text_all_ass_closed --> cases.

% text_all_ass_closed_tail is just like text_all_ass_closed, but can be empty.

text_all_ass_closed_tail --> statement, ['.'], text_all_ass_closed_tail.
text_all_ass_closed_tail --> definition, ['.'], text_all_ass_closed_tail.
text_all_ass_closed_tail --> assumption_text, text_all_ass_closed_tail.
text_all_ass_closed_tail --> cases_text, text_all_ass_closed_tail.
text_all_ass_closed_tail --> cases.
text_all_ass_closed_tail --> [].

% naproche_text_all_ass_closed is just like naproche_text, but without unclosed assumptions.

naproche_text_all_ass_closed --> axiom, naproche_text_all_ass_closed.
naproche_text_all_ass_closed --> lemma, naproche_text_all_ass_closed.
naproche_text_all_ass_closed --> theorem, naproche_text_all_ass_closed.
naproche_text_all_ass_closed --> assumption_naproche_text, naproche_text_all_ass_closed.
naproche_text_all_ass_closed --> text_all_ass_closed, naproche_text_all_ass_closed.
naproche_text_all_ass_closed --> [].

% closing starts with an assumption closing trigger.

closing --> ass_closing_trigger, proposition_coord(mode~finite), ['.'].

% CASES

% A cases text consists of cases and a case closing.

cases_text --> cases, case_closing.

```

```

% cases consists of an introductory sentence announcing N cases and a list of N cases.

cases --> [then], [there], [are], [N], [cases], [':'], ncases(N), {N >= 2}.
cases --> [now], [there], [are], [N], [cases], [':'], ncases(N), {N >= 2}.

% ncases(N) is a list of N cases"

ncases(0) --> [].
ncases(N) -->
{M is N-1, M >= 0},
ncases(M), ['case'], [N], [':'], proposition_coord(mode~finite), [':'], text.

% case_closing starts with case_clasing_trigger"

case_closing --> case_closing_trigger, proposition_coord(mode~finite), [':'].
case_closing --> statement_trigger, case_closing_trigger, proposition_coord(mode~finite), [':'].

% AXIOMS, LEMMAS AND THEOREMS

% An axiom consists of a heading followed by text.

axiom --> [axiom], [':'], proposition_coord(mode~finite), [':'].
axiom --> [axiom], identifier, [':'], proposition_coord(mode~finite), [':'].

% A lemma consists of a heading, a goal text, the marker "Proof", a body text and the marker "Qed".

lemma --> [lemma], [':'], text, [proof], [':'], text, [qed], [':'].
lemma --> [lemma], identifier, [':'], text, [proof], [':'], text, [qed], [':'].

% A lemma consists of a heading, a goal text, the marker "Proof", a body text (which possibly includes
% lemmas) and the marker "Qed".

theorem --> [theorem], [':'], text, [proof], [':'], theorem_text, [qed], [':'].
theorem --> [theorem], identifier, [':'], text, [proof], [':'], theorem_text, [qed], [':'].

% A theorem text is just like a text, but may contain lemmas.

theorem_text --> statement, [':'], theorem_text_tail.
theorem_text --> definition, [':'], theorem_text_tail.
theorem_text --> lemma, theorem_text_tail.
theorem_text --> assumption_text, theorem_text_tail.
theorem_text --> assumption, [':'], theorem_text_tail.
theorem_text --> cases_text, theorem_text_tail.
theorem_text --> cases.

% theorem_text_tail is just like theorem_text, but can be empty.

theorem_text_tail --> statement, [':'], theorem_text_tail.
theorem_text_tail --> definition, [':'], theorem_text_tail.
theorem_text_tail --> lemma, theorem_text_tail.
theorem_text_tail --> assumption_text, theorem_text_tail.
theorem_text_tail --> assumption, [':'], theorem_text_tail.
theorem_text_tail --> cases_text, theorem_text_tail.
theorem_text_tail --> cases.
theorem_text_tail --> [].

%=====
% proposition_coord
%=====

% proposition_coord can be either a simple sentence_coord, or a number of sentence_coords linked with
% "if...then" or "iff".
% The gulp feature "mode" can take three values: "that" indicates that the subclause is to be prefixed
% with "that", "infinitive" indicates that the verb of the sentence is in the infinitive, and "finite"
% is the normal case.
% The gulp feature "that" defines whether the proposition_coord is a clause subordinated with a "that".
% It can tak values "plus" or "minus".

proposition_coord(mode~Mode) -->

```



```

sentence_coord(mode~Mode).
proposition_coord(mode~finite) -->
[if], sentence_coord(mode~finite), [then], sentence_coord(mode~finite).
proposition_coord(mode~that) -->
[that], [if], sentence_coord(mode~finite), [then], sentence_coord(mode~finite).
proposition_coord(mode~finite) -->
sentence_coord(mode~finite), iff, sentence_coord(mode~finite).
proposition_coord(mode~infinitive) -->
sentence_coord(mode~infinitive), iff, sentence_coord(mode~finite).
proposition_coord(mode~finite) -->
sentence_coord(mode~finite), [if], sentence_coord(mode~finite).
proposition_coord(mode~infinitive) -->
sentence_coord(mode~infinitive), [if], sentence_coord(mode~finite).

iff --> [iff].
iff --> [if], [and], [only], [if].

%=====
% sentence_coord
%=====

% sentence_coord works like in Attempo (with "i.e." added). It links a number of topicalised_sentences
% with "and", "or", " ", and", " ", or" and "i.e." in such a way that the bracketing is unambiguous.
% No coordination is allowed in infinitive mode.

sentence_coord(mode~infinitive) --> !, composite_sentence(mode~infinitive).

sentence_coord(mode~Mode) --> sentence_coord_0(mode~Mode), sentence_coord_tail(mode~Mode).
sentence_coord_tail(mode~Mode) --> [' ','i.e.'], sentence_coord(mode~Mode).
sentence_coord_tail(mode~_) --> [].

sentence_coord_0(mode~Mode) --> sentence_coord_1(mode~Mode), sentence_coord_0_tail(mode~Mode).
sentence_coord_0_tail(mode~Mode) --> [' ','or'], sentence_coord_0(mode~Mode).
sentence_coord_0_tail(mode~_) --> [].

sentence_coord_1(mode~Mode) --> sentence_coord_2(mode~Mode), sentence_coord_1_tail(mode~Mode).
sentence_coord_1_tail(mode~Mode) --> [' ','and'], optional_trigger, sentence_coord_1(mode~Mode).
sentence_coord_1_tail(mode~_) --> [].

sentence_coord_2(mode~Mode) --> sentence_coord_3(mode~Mode), sentence_coord_2_tail(mode~Mode).
sentence_coord_2_tail(mode~Mode) --> ['or'], sentence_coord_2(mode~Mode).
sentence_coord_2_tail(mode~_) --> [].

sentence_coord_3(mode~finite) --> topicalised_sentence, sentence_coord_3_tail(mode~finite).
sentence_coord_3(mode~that) --> [that], topicalised_sentence, sentence_coord_3_tail(mode~that).
sentence_coord_3_tail(mode~Mode) --> [and], optional_trigger, sentence_coord_3(mode~Mode).
sentence_coord_3_tail(mode~_) --> [].

% A topicalised_sentence can be a simple composite_sentence, or a quantified statement.

topicalised_sentence --> existential_topic.
topicalised_sentence --> universal_topic, proposition_coord(mode~finite).
topicalised_sentence --> composite_sentence(mode~finite).

existential_topic --> exist_quantifier, np.
existential_topic --> exist_quantifier, variable_list_bar.

universal_topic --> universal_quantifier, nbar.
universal_topic --> universal_quantifier, variable_list_bar.
% The following is needed as long as plural is not implemented properly:
universal_topic --> [for], [all], variable_list_bar, optional_comma.

variable_list_bar --> variable_list, optional_comma, such_that_clause.
variable_list_bar --> variable_list.

variable_list --> [math(X)], variable_list_tail, {fo_variable(_,X,[])}.
variable_list_tail --> optional_comma, [math(X)], variable_list_tail, {fo_variable(_,X,[])}.
variable_list_tail --> [].

```

```

such_that_clause --> [such], [that], topicalised_sentence.

% A composite_sentence is either a simple sentence, a formula or a proposition_coord prefixed
% with a sentence_init. By the constructions that contain composite_sentence, it is given that
% the "mode" feature can here only take the values "finite" and "infinitive" (not "that").

composite_sentence(mode~Mode) --> sentence(mode~Mode).
composite_sentence(mode~finite) --> sentence_init, proposition_coord(mode~that).
composite_sentence(mode~_) --> [math(X)], {fo_formula(_,_,X,[])}

sentence_init --> [it], [is], [false].
sentence_init --> [it], [is], [not], [the], [case].
sentence_init --> [it], [is], [the], [case].

% A sentence is just a noun phrase followed by a verb phrase"

sentence(mode~finite) --> np, vp(infinitive~minus).
sentence(mode~infinitive) --> np, vp(infinitive~plus).

%=====
% Noun phrases
%=====

np --> [math(X)], {fo_term(_,_,X,[])}
np --> pronoun.
np --> specifier, nbar.

nbar --> adjective, nbar1.
nbar --> nbar1.
nbar1 --> noun, variablebar.
nbar1 --> noun.
nbar1 --> variablebar.

variablebar --> [math(X)], such_that_clause, {fo_variable(_,X,[])}
variablebar --> [math(X)], {fo_variable(_,X,[])}

specifier --> indefinite_specifier.
specifier --> definite_specifier.
specifier --> universal_specifier.
specifier --> numerical_specifier.
specifier --> negative_specifier.

%=====
% Verb phrases
%=====

vp(infinitive~Infinitive) --> vbar(infinitive~Infinitive).

vp(infinitive~minus) --> [does], [not], vbar(infinitive~plus).
vp(infinitive~plus) --> [not], vbar(infinitive~plus).

vp(infinitive~minus) --> [is], adjective.
vp(infinitive~minus) --> [is], [not], adjective.
vp(infinitive~minus) --> [is], indefinite_specifier, nbar.
vp(infinitive~minus) --> [is], [not], indefinite_specifier, nbar.

vp(infinitive~plus) --> [be], adjective.
vp(infinitive~plus) --> [not], [be], adjective.
vp(infinitive~plus) --> [be], indefinite_specifier, nbar.
vp(infinitive~plus) --> [not], [be], indefinite_specifier, nbar.

vbar(infinitive~Infinitive) --> verb(transitive~plus..infinitive~Infinitive), np.
vbar(infinitive~Infinitive) --> verb(transitive~minus..infinitive~Infinitive).

%=====
% Lexicon
%=====

```

```
% NOUNS AND PRONOUNS

noun --> [set].
noun --> [element].
noun --> [number].
noun --> [integer].

pronoun --> [it].

% ADJECTIVES AND VERBS

adjective --> [empty].
adjective --> [even].
adjective --> [odd].
adjective --> [prime].
adjective --> [positive].
adjective --> [transitive].
adjective --> [square].

verb(transitive~plus..infinitive~minus) --> [contains].
verb(transitive~plus..infinitive~plus) --> [contain].
verb(transitive~plus..infinitive~minus) --> [divides].
verb(transitive~plus..infinitive~plus) --> [divide].

% SPECIFIERS AND QUANTIFIERS

indefinite_specifier --> [a].
indefinite_specifier --> [an].
indefinite_specifier --> [some].

definite_specifier --> [the].

numerical_specifier --> [precisely], [one].

negative_specifier --> [no].

universal_specifier --> [every].

universal_quantifier --> [for], [every].
% universal_quantifier --> [for], [all].

exist_quantifier --> [there], [is].
exist_quantifier --> [there], [exists].

% TRIGGERS

statement_trigger --> [then].
statement_trigger --> [hence].
statement_trigger --> [therefore].
statement_trigger --> [recall], [that].
statement_trigger --> [but].
statement_trigger --> [in], [particular].
statement_trigger --> [observe], [that].
statement_trigger --> [together], [we], [have].
statement_trigger --> [so].

ass_trigger --> [assume].
ass_trigger --> [suppose].
ass_trigger --> [assume], [that].
ass_trigger --> [suppose], [that].
ass_trigger --> [assume], [for], [a], [contradiction], [that].

ass_closing_trigger --> [thus].

case_closing_trigger --> [in], [all], [cases], optional_comma.
case_closing_trigger --> [in], [both], [cases], optional_comma.
```

```
% OTHER STUFF

optional_comma --> [','].
optional_comma --> [].

optional_trigger --> statement_trigger.
optional_trigger --> [].

identifier --> [..].
```

---

# Literaturverzeichnis

---

- [1] P. Blackburn and J. Bos. Representation and Inference for Natural Language: Working with Discourse Representation Structures. 1999.
- [2] M. A. Covington. GULP 3.1: An Extension of Prolog for Unification-Based Grammar. 1994.
- [3] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Einführung in die mathematische Logik*. Spektrum, 2007.
- [4] G. Frege. *Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. 1879.
- [5] N. E. Fuchs, S. Höfler, K. Kaljurand, F. Rinaldi, and G. Schneider. *Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines*.
- [6] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers (Fourth Edition)*. 1960.
- [7] H. Kamp and U. Reyle. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language*. Kluwer Academic Publisher, 1993.
- [8] N. Kolev. Generating Proof Representation Structures for the Project NAPROCHE. Master's thesis, University of Bonn, 2008.
- [9] D. Kühlwein. Ein Kalkül für Proof Representation Structures, Diplomarbeit, 2008.
- [10] R. Matuszewski and P. Rudnicki. Mizar: the first 30 years. *Mechanized Mathematics and Its Applications*, 4:2005, 2005.
- [11] A. N. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1962.