

This version: April 6, 2009

# An experimental grammar for German mathematical text

**Peter Schodl**  
**Arnold Neumaier**

*Fakultät für Mathematik, Universität Wien*  
*Nordbergstr. 15, A-1090 Wien, Austria*  
*email: Peter.Schodl@univie.ac.at*  
*email: Arnold.Neumaier@univie.ac.at*  
*WWW: <http://www.mat.univie.ac.at/~neum/>*

## **Abstract**

This is a report on work in progress, for the FMathL project. It describes a preliminary grammar designed for ultimately parsing the German language lecture notes on analysis and linear algebra by the second author. The report will be updated from time to time. The most recent version of this paper is always available at <http://www.mat.univie.ac.at/~neum/FMathL.html#ALA>.

# Contents

<b>I Overview</b>	<b>3</b>
1 Introduction	3
2 Morphology	4
3 Syntax	9
<b>II The lexicon</b>	<b>9</b>
4 Word lexicon	9
5 Endings lexicon	20
6 Phrase lexicon	20
<b>III The grammar</b>	<b>21</b>
7 Lexical productions	22
8 Global productions	24
9 Definition productions	25
10 Statement productions	28
11 Proof productions	31
<b>IV Bibliography</b>	<b>45</b>

## Part I

# Overview

## 1 Introduction

This is a report on work in progress, for the FMathL project. It describes a preliminary grammar designed for ultimately parsing the German language lecture notes on analysis and linear algebra by the second author. The report will be updated from time to time. The most recent version of this paper is always available at [2].

We give a preliminary grammar in Backus-Naur-form (BNF) of a language that should, after further development, allow to parse mathematics close to natural language. For a definition of the Backus-Naur-form, see COHEN [3].

Our goal was to be able to make sense out of mathematical text given as a  $\text{\LaTeX}$ -file, at first without considering formulas in the text. We emphasize that we are interested in *mathematical* text which essentially represents the logical relations between well-defined mathematical objects. Hence the language is much smaller and simpler than general natural language. We do not need to understand metaphors, jokes etc., or talk about the language itself – some of the things that make general natural language so hard to understand for machines.

Since the language of this grammar is restricted, but still sounds like natural language, it is a special case of a **controlled natural language** (CNL). There already exist several controlled natural languages, e.g., the ATTEMPTO project [8] at the University of Zürich, which does not have a specific mathematical emphasis. Another CNL with a strong emphasis on proof checking is used by the NAPROCHE project [6] at the University of Bonn.

More specifically, our grammar is designed to parse one specific text, and will be tested and optimized for this text. The text under consideration is a set of lecture notes in German on a beginner's course on analysis and linear algebra by NEUMAIER [1], about 450 pages in length. We chose this text because it starts with higher mathematics from scratch, hence almost everything is explicitly defined, and due to its length it should provide a rather expressive grammar that contains a wide variety of basic constructions of mathematical language.

As a reasonable compromise between flexibility and accurateness, we selected a grammar intended to be unambiguous concerning the logical relation represented by each production, i.e., two sentences with different logical structure will not be parsed by the same production. A complete list of the productions and their interpretations, where already rendered, is given in Part III.

The text to be parsed is assumed to be already divided into sentences, for which the environment is known, i.e., we parse sentences which are known to be either sentences in a theorem, a definition, a proof, comment etc. Everything in a  $\text{\LaTeX}$  formula or display environment is replaced by the string `FORMULA` and `DISPLAY` respectively, its contents is saved for separate analysis. Similarly, we replaced arbitrary references by `REF` and ellipsis by `DOTS`. We made further simplifications: all German umlauts have been replaced by their ASCII-equivalent. At present, we do not distinguish between upper and lower case letters, hence every sentence is converted to lower case. The list of resulting **ALA-sentences** is available at [2].

Parsing an ALA-sentence is done in three steps: Beginning on the word-level, the grammar produces different grammatical forms from words in the **lexicon**, i.e., a database of words

together with the information, to which category it belongs. These productions are called the **morphology**. The lexicon consists of the **word lexicon**, containing 1426 plain words, the **phrase lexicon**, containing 56 phrases, and the **ending lexicon**, which contains 40 endings for inflections. The entries of the lexicon are given in Part II.

For example, the morphology produces the words **negative**, **negatives**, **negativer** etc. from the word lexicon-entry **negativ** and the endings lexicon entries **e**, **es** and **er** respectively. Then, phrases are produced by the grammar, for example **vektorraum** produces **der vektorraum**, **beueglich vektorraum FORMULA**, **vektorraum ueber FORMULA** etc. In the last step, phrases are combined to sentences, e.g., the phrases **fuer FORMULA gilt FORMULA** and also **ist FORMULA eine gruppe** are combined to the sentence **fuer FORMULA gilt FORMULA also ist FORMULA eine gruppe**.

Mathematical notions are replaced by **categories**, i.e., a token that indicates a set of words with similar meaning. Categories are printed in the grammar as words prefixed with single or double quotes. For example **also**, **daher**, **deswegen** etc. which all stand for an implication and are found in the category **"hence**.

## 2 Morphology

This section treats the way our grammar produces the grammatical forms out of some root-form of a word. Ultimately, the Grammatical Framework (GF) [5], a project at the University of Göteborg, will take care of this task. The Grammatical Framework is multi-lingual, see e.g., ANGELOV [9].

At the moment, we approximate a fully developed morphological analysis by a small **morphological grammar**. This is a first-stage grammar independent of the main grammar. The morphologic grammar has its own categories, all beginning with a single quotation mark. A complete list of these categories is given in Table 1.

The morphologic grammar allows to inflect words, but also to combine words, e.g. **funktionen** and **raum** can be combined to **funktionenraum**. Furthermore, it can produce variants of a word, e.g., from **definit** it can produce the words **indefinit**, **semidefinit** etc. Note that the morphological grammar is quite liberal, i.e., it is intended to recognize a *superset* of the set of words in ALA, and generates many words that would not be proper German.

The morphological grammar only concatenates words without blanks, hence it is guaranteed that no word produced by the morphological grammar contains a blank. Lexicon entries to be added to the list of productions are indicated by **&words** (for entries in the word lexicon) and **&endings** (for entries in the ending lexicon).

The current morphological grammar is as follows. (As usual, a pipe | separates alternatives for a nonterminal to be replaced.)

**Productions for 'o** (object, either 'do or 'udo)

```
'o = 'do | 'udo | 'n'o | 'o'o | 'p'o | 'o-'o | 'o'op | 'p'pt | 'v-'o
    | 'o'op'o
```

**Productions for 'n** (name modifiers)

```
'n = &words | 'n'op | 'n'nt'op
```

Category	Description
'o	object, either 'do or 'udo
'n	name modifiers
'art	articles
'p	property (distinct, unique, semidefinite, ...)
'pam	prefixes for 'am
'aa	actions that algorithms ("we") take (multiply, cancel, reorder, ...)
'am	actions that objects take (converge, contradict, exist, ...)
'qt	quantification (some, any, $n$ , ...)
'rg	reference outside the context
'ad	endings for 'art
'amt	ending for the use of an action as property
'ap	endings for plural forms of 'am and 'aa
'because	variations of "because"
'but	variations of "but"
'dash	dashes
'dir	directions
'dird	endings for 'dir
'do	defined object (mapping, root, approximation, ...)
'dots	ellipsis
'empty	the empty string
'f	formula
'fd	formula designator (equation, equality, expression, ...)
'follows	variations of "follows"
'hence	variations of "hence"
'if	variations of "if"
'nr	ordinal numbers
'nt	name termination
'op	endings for plural forms of 'do and 'udo
'pm	prefixes for 'p
'pp	postfixes for plural forms of 'p
'proofline	one-word proofs (trivial, analogous, ...)
'prooflineadd	one-word additional information
'pt	postfixes for nominal use of 'p
'punct	punctuation
'qd	for inflections of 'qt
'rd	reference designator (theorem, lemma, example, ...)
'rl	reference to something in the same context
'since	variations of "since" or "by"
'suppose	variations of "suppose"
'te	endings for enumerations
'udo	undefined object (property, criterion, concept, ...)
'v	variables

Table 1: Morphological categories

**Productions for 'art** (articles)

'art = &words | 'art'ad

**Productions for 'p** (property)

'p = &words | 'p'p | 'n-'p | 'pm'p | 'p'pp | 'v'te | 'am'amt

**Productions for 'pam** (prefixes for 'am)

'pam = &words | 'empty

**Productions for 'aa** (actions that algorithms (“we”) take)

'aa = &words | 'aa'ap | 'pam'aa

**Productions for 'am** (actions that objects take)

'am = &words | 'am'ap | 'pam'am

**Productions for 'qt** (quantification)

'qt = &words | 'qt'qd

**Productions for 'rg** (reference outside the context)

'rg = &words | 'n'rd | 'o'rd | 'n-'rd | 'o-'rd

**Productions for 'ad** (endings for 'art)

'ad = &endings | 'empty

**Productions for 'amt** (ending for the use of an action as property)

'amt = &endings

**Productions for 'ap** (endings for plural forms of 'am and 'aa)

'ap = &endings | 'empty

**Productions for 'because** (variations of “because”)

'because = &words

**Productions for 'but** (variations of “but”)

'but = &words

**Productions for 'dash** (dashes)

'dash = &words

**Productions for 'dir** (directions)

'dir = &words

**Productions for 'dird** (endings for 'dir)

'dird = &words | 'empty

**Productions for 'do** (defined object)

'do = &words

**Productions for 'dots** (ellipsis)

'dots = &words

**Productions for 'empty** (the empty string)

'empty =

**Productions for 'f** (formula)

'f = &words

**Productions for 'fd** (formula designator)

'fd = &words

**Productions for 'follows** (variations of “follows”)

'follows = &words

**Productions for 'hence** (variations of “hence”)

'hence = &words

**Productions for 'if** (variations of “if”)

'if = &words

**Productions for 'nr** (ordinal numbers)

'nr = &words

**Productions for 'nt** (name termination)

'nt = &words

**Productions for 'op** (endings for plural forms of 'do and 'udo)

'op = &endings | 'empty

**Productions for 'pm** (prefixes for 'p)

'pm = &words

**Productions for 'pp** (postfixes for plural forms of 'p)

'pp = &endings | 'empty

**Productions for 'proofline** (one-word proofs)

'proofline = &words

**Productions for 'prooflineadd** (one-word additional information)

'prooflineadd = &words

**Productions for 'pt** (postfixes for nominal use of 'p)

'pt = &words

**Productions for 'punct** (punctuation)

'punct = &words

**Productions for 'qd** (for inflections of 'qt)

'qd = &endings | 'empty

**Productions for 'rd** (reference designator)

'rd = &words

**Productions for 'rl** (reference to something in the same context)

'rl = &words

**Productions for 'since** (variations of "since" or "by")

'since = &words

**Productions for 'suppose** (variations of “suppose”)

'suppose = &words

**Productions for 'te** (endings for enumerations)

'te = &endings

**Productions for 'udo** (undefined object)

'udo = &words

**Productions for 'v** (variables)

'v = &words

### 3 Syntax

Now we are able to produce a wide variety of words in the first-stage grammar. But to gain more flexibility, we want to allow also groups of words to be of the same category, i.e., we firstly want to allow a phrase like **es folgt** to be in the same category as **folglich**, and secondly we want a words, that together make up the same kind of grammatical object as a single word, to be in the same category. For example, **nicht stetig** and **stetig** are both properties. We call these categories **lexical categories**, and they are given in Table 2. All lexical categories begin with a double quotation mark. Lexicon entries to be added to the list of phrases are indicated by **&phrases** (for entries in the phrases lexicon).

As a last step is to put these phrases together to sentences, for example a definition is made up from a phrase that introduce a name to be defined, and a second phrase that actually gives the definition. For example, to parse the sentence **Wir nennen A positiv definit, wenn FORMULA gilt**, we could replace the part preceding the comma by **A heisst positiv definit**, without altering the meaning. Hence we introduced a category for those parts of a sentence. We did similar things for sentences which express an implication, an equivalence, etc. The productions on the sentence-level deal with the **syntactical categories**, given in Table 3.

The grammar has 1070 productions, given in Part III. On the other hand, we do *not* use the traditional grammatical categories (verb, noun, etc.).

## Part II

# The lexicon

### 4 Word lexicon

,:	'punct	--:	'dash	..:	'punct
-:	'dash	---:	'dash	:::	'punct

Category	Description
"aa	actions that algorithms (“we”) take (multiply, cancel, reorder, ...)
"am	actions that objects take (converge, contradict, exist, ...)
"but	variations of “but”
"dir	direction (left, above, ...)
"f	formula
"follows	variations of “follows”
"hence	variations of “hence”
"if	variations of “if”
"n	name modifiers (hesse-, principal-, eigen-, ...)
"o	object, either ’do or ’udo
"p	property (distinct, unique, semidefinite, ...)
"ql	qualification of ’am and ’p (in point, relative to, ...)
"qt	quantification (some, any, $n$ , ...)
"r	reference
"rg	reference outside the context
"rl	reference to something in the same context
"since	variations of “since” or “by”
"suppose	variations of “suppose”
"v	variable

Table 2: Lexical categories

Category	Description
"sentence	complete sentence
"context	a sentence that establishes some context
"defcomment	some comment to a definition
"statement	statement
"statementadd	some additional information to a statement
"def1left	the first part of a definition of type 1
"def1right	the second part of a definition of type 1
"def2left	the first part of a definition of type 2
"def2right	the second part of a definition of type 2
"defsentence	a sentence that defines something
"equileft	the first part of an equivalence
"equiright	the second part of an equivalence
"implicleft	the first part of an implication
"implicright	the second part of an implication
"proofline	a sentence in a proof
"prooflineadd	some additional information to a sentence in a proof

Table 3: Syntactical categories

;;	'punct	adisch:	'p	anzahl:	'do
DISPLAY:	'f	adjungiert:	'p	approximation:	'do
DOTS:	'dots	adjunkt:	'p	approximier:	'am
EQN:	'f	aehnlich:	'p	approximierbar:	'p
FORMULA:	'f	aender:	'am	arbeit:	'do
FORMULA:	'v	aequidistant:	'p	argument:	'do
PREFIX:	'v	aequivalent:	'p	argument:	'udo
REF:	'rg	aequivalenz:	'do	arithmetisch:	'p
ab:	'pam	aeusser:	'p	art:	'udo
abaender:	'am	aeussere:	'p	assoziativ:	'p
abbild:	'do	affin:	'p	assoziativgesetz:	'rg
abbilde:	'am	algebra:	'do	asymptotisch:	'p
abbildung:	'do	algebraisch:	'p	atlanten:	'do
abbildungsgrad:	'do	algebren:	'do	atlas:	'do
abbrech:	'am	all:	'qt	auf:	'pam
abelsch:	'p	allgemein:	'p	aufgeschlitzt:	'p
aber:	'but	allgemeiner:	'p	aufgeschlitzte:	'p
abgebildet:	'p	allgemeinfall:	'udo	aufgespannt:	'p
abgebrochen:	'p	also:	'follows	aufloes:	'aa
abgeleitet:	'p	also:	'hence	aufloes:	'am
abgeschaetzt:	'p	alternier:	'am	aufloesbar:	'p
abgeschlossen:	'p	alternierend:	'p	aufschlitz:	'aa
abgeschnitten:	'p	amplitude:	'do	aufspalt:	'aa
abhaeng:	'am	analog:	'proofline	aufspann:	'am
abhaengig:	'p	analog:	'p	aufsteigend:	'p
abkuerzung:	'fd	analyse:	'do	aufsummier:	'aa
ableit:	'aa	analyse:	'udo	auftret:	'am
ableitung:	'do	analysis:	'do	auftretend:	'p
abschaetz:	'aa	analytisch:	'p	auftritt:	'am
abschaetz:	'am	anfang:	'do	aufzuloesen:	'p
abschaetzung:	'do	anfangsbedingung:	'do	aus:	'since
abschaetzung:	'fd	angabe:	'udo	ausdruck:	'do
abschaetzung:	'rd	angemessen:	'p	ausdruck:	'udo
abschluss:	'do	angenommen:	'if	ausdruecke:	'do
abschwaechung:	'do	angenommen:	'suppose	ausfuehrung:	'udo
absolut:	'p	annahme:	'rd	ausklammer:	'aa
abstaende:	'do	annehm:	'am	auslenkung:	'do
abstand:	'do	annimmt:	'am	ausmultiplizier:	'aa
abstandserhaltend:	'p	anordnung:	'udo	ausnahme:	'do
absteigend:	'p	anpassung:	'do	ausrechn:	'aa
abweichung:	'do	anregung:	'do	ausreich:	'am
abzaehlbar:	'p	ansatz:	'udo	ausreichend:	'p
abzaehlung:	'do	ansteig:	'am	aussage:	'rd
abzuleiten:	'p	anstieg:	'do	aussage:	'udo
abzuschuetzen:	'p	anti:	'pm	aussagen:	'do
achse:	'do	antikommutativ:	'p	ausschreib:	'aa
addier:	'aa	anwend:	'aa	auswaehl:	'aa
additionstheorem:	'do	anwendbar:	'p	auszuwaehlen:	'p
additiv:	'p	anwendung:	'do	auszuwerten:	'p

automorphismen: 'do	beschreibend: 'p	brouwer: 'n
automorphismus: 'do	beschreibung: 'udo	bruch: 'do
axiom: 'do	beschrieben: 'p	buchstabe: 'udo
axiom: 'rd	besitz: 'am	cauchy: 'n
axiomatisch: 'p	besondere: 'p	cavalieri: 'n
axiome: 'rd	best: 'p	cayley: 'n
banach: 'n	besteh: 'am	charakterisiert: 'p
banachschen: 'n	bestimmt: 'p	charakteristisch: 'p
band: 'do	betraege: 'do	cholesky: 'n
barner: 'n	betrag: 'do	civita: 'n
basen: 'do	betragmaessig: 'p	codierung: 'do
basis: 'do	bewegung: 'do	cosinus: 'do
bedingt: 'p	beweis: 'aa	cramer: 'n
bedingung: 'do	beweis: 'rd	d.h.: 'hence
bedingung: 'fd	beweis: 'udo	da: 'because
bedingung: 'rd	beweisbar: 'p	dach: 'do
beend: 'aa	beweisend: 'p	daempfung: 'do
beginn: 'aa	bewiesen: 'p	daher: 'follows
begriff: 'udo	bezeichn: 'aa	daher: 'hence
behauptet: 'p	bezeichnung: 'udo	damit: 'follows
behauptung: 'do	bezieh: 'aa	damit: 'hence
behauptung: 'rd	beziehung: 'do	daniell: 'n
behauptungen: 'do	beziehung: 'fd	dann: 'follows
beid: 'qt	beziehung: 'rd	dann: 'hence
beide: 'qt	beziehung: 'udo	darstell: 'aa
beidesmal: 'qt	bezug: 'do	darstell: 'am
beinhalte: 'am	bijektion: 'do	darstellbar: 'p
beispiel: 'rd	bijektiv: 'p	darstellung: 'udo
beispiel: 'udo	bild: 'aa	das: 'art
beispielsweise: 'p	bild: 'am	das: 'rl
beitraeg: 'am	bild: 'do	dasjenig: 'art
beitrag: 'am	bilde: 'aa	dass: 'follows
bekannt: 'p	bilde: 'am	dasselbe: 'p
beliebig: 'p	bildung: 'do	daten: 'do
benachbart: 'p	bildung: 'udo	dazuaddier: 'aa
benoetig: 'aa	bilinear: 'p	defektiv: 'p
benoetigt: 'p	binaer: 'p	definier: 'am
benutz: 'aa	binomial: 'p	definiert: 'p
berechne: 'am	binomisch: 'p	definit: 'p
berechnung: 'udo	block: 'do	definition: 'do
bereich: 'do	blockdiagonal: 'p	definition: 'rd
bernoulli: 'n	bloecke: 'do	delta: 'do
beruh: 'am	bogen: 'do	dem: 'art
besag: 'am	bolzano: 'n	dem: 'rl
beschleunigung: 'do	borel: 'n	den: 'art
beschraenk: 'am	bosonen: 'n	denn: 'because
beschraenkt: 'p	bra: 'do	der: 'art
beschreib: 'am	bricht: 'am	derart: 'prooflineadd
beschreibbar: 'p	bronstein: 'n	derartig: 'art

derjenig: 'art	dritten: 'nr	en: 'dird
derselbe: 'p	dual: 'do	end: 'n
des: 'art	durch: 'since	endlich: 'p
deshalb: 'hence	durchgang: 'do	endomorphismus: 'do
dessen: 'art	durchlaeuf: 'am	energie: 'do
deswegen: 'hence	durchlauf: 'am	ensemble: 'do
deswegen: 'rl	durchlauf: 'do	entartet: 'p
determinante: 'do	durchschnitt: 'do	entbehrlich: 'p
dezimal: 'do	e: 'dird	entfernen: 'am
dezimal: 'p	ebene: 'do	entgegengesetzt: 'p
diagonal: 'p	ebenso: 'proofline	enthaelt: 'am
diagonale: 'do	ebensolche: 'art	enthalt: 'am
diagonalisierbar: 'p	echt: 'p	enthalten: 'p
diagramm: 'do	eckig: 'p	enthaltend: 'p
dicht: 'p	effekt: 'do	enthielt: 'am
dichte: 'do	effizient: 'p	entropie: 'do
die: 'art	eigen: 'n	entsprech: 'am
diejenig: 'art	eigenschaft: 'fd	entsprechend: 'p
dies: 'art	eigenschaft: 'rd	entsteh: 'am
dies: 'rl	eigenschaft: 'udo	entstehend: 'p
diese: 'rl	eigenschaften: 'rd	entwickel: 'aa
dieselbe: 'p	eigenvektor: 'do	entwicklung: 'do
differential: 'do	eigenwert: 'do	epimorphismus: 'do
differenz: 'do	ein: 'art	er: 'dird
differenzierbar: 'p	ein: 'qt	erfuell: 'am
dimension: 'do	eindeutig: 'p	ergaenz: 'aa
dimensionsformel: 'rg	einfach: 'p	ergaenzung: 'udo
direkt: 'p	einfacher: 'p	ergeb: 'am
dirichlet: 'n	eingeschraenkt: 'p	ergebnis: 'udo
disjunkt: 'p	einheit: 'do	ergibt: 'am
diskret: 'p	einheit: 'n	ergibt: 'follows
distributionen: 'do	einig: 'qt	erhalten: 'follows
distributiv: 'p	einmal: 'qt	erheblich: 'p
divergent: 'p	eins: 'do	ersetzt: 'aa
divergenz: 'do	einschraenkung: 'do	erst: 'p
divergier: 'am	einseitig: 'p	erste: 'nr
divergierend: 'p	einsen: 'do	ersten: 'nr
dividier: 'aa	einsetz: 'aa	erwartungswert: 'do
dividier: 'am	einspaltig: 'p	erweitern: 'am
dominant: 'p	einstein: 'n	erweitert: 'p
doppel: 'n	einzeilig: 'p	erweiterung: 'do
doppeldeutig: 'p	einzel: 'p	erzeug: 'aa
doppelt: 'p	einzig: 'p	erzwungen: 'p
drehung: 'do	elektron: 'do	es: 'dird
drei: 'qt	element: 'do	euklidisch: 'p
dreieck: 'do	elementar: 'p	euler: 'n
dreieckig: 'p	elimination: 'do	exakt: 'p
dritt: 'p	ellipse: 'do	exemplarisch: 'p
dritte: 'nr	ellipsoid: 'do	existent: 'p

existier: 'am	formel: 'rd	gesetz: 'do
experiment: 'udo	formulierung: 'do	gesetz: 'rd
explizit: 'p	forster: 'n	gesetze: 'do
exponent: 'do	fortsetzung: 'do	gesetze: 'rd
exponential: 'n	fourier: 'n	gewisse: 'p
exponentiell: 'p	frequenz: 'do	gilt: 'am
extremum: 'do	fubini: 'n	gilt: 'follows
extremwert: 'do	fuehrend: 'p	gilt: 'if
faktor: 'do	fuer: 'since	glatt: 'p
faktorisier: 'aa	fundamental: 'n	gleich: 'p
faktorisieren: 'do	fundamental: 'p	gleichbedeutend: 'p
faktorisierung: 'do	fundamentalsatz: 'rg	gleichgewicht: 'do
fakultaet: 'do	funktion: 'do	gleichheit: 'fd
fallend: 'p	funktional: 'do	gleichmaessig: 'p
falls: 'if	galilei: 'n	gleichung: 'do
falsch: 'p	gamma: 'do	gleichung: 'fd
familie: 'do	ganz: 'p	gleichung: 'rd
fatou: 'n	ganzzahlig: 'p	gleichung: 'udo
fehl: 'am	gauss: 'n	gleichungen: 'fd
fehlend: 'p	gebiet: 'do	gleichungen: 'rd
fehler: 'do	gedaempft: 'p	gleichungssystem: 'fd
fehlerbehaftet: 'p	geeignet: 'p	gleichviele: 'qt
feld: 'do	gefordert: 'p	gleichwertig: 'p
feldlinie: 'do	gegebenenfalls: 'if	gleichzeitig: 'p
fest: 'p	gegeignet: 'p	glied: 'do
festgeleg: 'am	gegenbeispiel: 'do	glied: 'udo
festgelegt: 'p	gegenbeispiel: 'udo	gliedweise: 'p
festlaesst: 'am	gegenseitig: 'p	global: 'p
figur: 'do	geh: 'am	grad: 'do
finden: 'follows	gehoer: 'am	gradient: 'do
finit: 'p	gekoppelt: 'p	gram: 'n
fixpunkt: 'do	gelt: 'am	graph: 'do
fixpunktsatz: 'rg	gelte: 'suppose	gravitation: 'do
flaechen: 'do	gelten: 'follows	grenz: 'do
fletcher: 'n	genaehert: 'p	grenze: 'do
folg: 'am	genau: 'p	grenzfall: 'udo
folge: 'do	genauer: 'p	grenzwert: 'do
folgen: 'follows	genueg: 'am	groesse: 'do
folgend: 'dir	genuegend: 'p	groessenordnung: 'do
folgend: 'p	geometrie: 'do	groesser: 'p
folgerung: 'do	geometrisch: 'p	groesst: 'p
folgich: 'hence	geordnet: 'p	groesstmoeglich: 'p
folgt: 'follows	gerade: 'p	gross: 'p
forderung: 'fd	gesamt: 'n	grund: 'n
forderung: 'rd	gesamt: 'p	gruppe: 'do
forderung: 'udo	gesamtheit: 'do	gueltig: 'p
form: 'do	geschachtelt: 'p	hab: 'am
form: 'udo	geschlossen: 'p	habe: 'suppose
formel: 'do	geschwindigkeit: 'do	haeufig: 'qt

haeufungspunkt: 'do	idealisiert: 'p	irgendwo: 'qt
halboffen: 'p	ident: 'p	irrational: 'p
hamilton: 'n	identifikation: 'do	irregulaer: 'p
harmonisch: 'p	identisch: 'p	isoliert: 'p
hat: 'am	ihr: 'art	isometrisch: 'p
hatt: 'am	imaginaer: 'p	isomorphie: 'do
haupt: 'n	immer: 'qt	isomorphismen: 'do
hauptsatz: 'rg	implizieren: 'follows	isomorphismus: 'do
hausdorff: 'n	impliziert: 'follows	ist: 'am
heaviside: 'n	implizit: 'p	ist: 'follows
heine: 'n	in: 'pm	ist: 'if
herausheb: 'am	index: 'do	ist: 'since
herausziehe: 'aa	indizes: 'do	itaet: 'pt
hergeleitet: 'p	indiziert: 'p	iteration: 'do
herleitung: 'do	indizierung: 'do	jacobi: 'n
herleitung: 'rd	induktion: 'do	jed: 'qt
hermite: 'n	induktion: 'n	jedesmal: 'qt
hermitesch: 'p	induktion: 'udo	jedoch: 'but
hervorgeh: 'am	induktiv: 'proofline	jene: 'art
hesse: 'n	induktiv: 'p	jordan: 'n
hilbertraum: 'n	induziert: 'p	kanonisch: 'p
hilfs: 'n	ineinander: 'p	kardinalitaet: 'do
hilfssatz: 'rd	infimum: 'do	karte: 'do
hinten: 'p	infinitesimal: 'p	kartesisch: 'p
hinzufueg: 'aa	information: 'do	kegel: 'do
hinzunehm: 'aa	inhalt: 'do	kein: 'art
hochdimensional: 'p	inhomogen: 'p	kein: 'qt
hoechst: 'p	injektiv: 'p	keinerlei: 'qt
hoehenlinie: 'do	inkommensurabel: 'p	keit: 'pt
hoeher: 'p	innen: 'do	ket: 'do
holomorph: 'p	innen: 'p	kette: 'do
homeomorph: 'p	inner: 'do	kettenregel: 'rg
homeomorphismus: 'do	inneres: 'do	kinetisch: 'p
homoeomorph: 'p	innerhalb: 'p	klammern: 'do
homoeomorphismen: 'do	integral: 'do	klar: 'proofline
homoeomorphismus: 'do	integrand: 'do	klar: 'p
homogen: 'p	integration: 'do	klasse: 'do
homomorphismen: 'do	integrierbar: 'p	klassisch: 'p
homomorphismus: 'do	integritaet: 'do	klein: 'p
homotop: 'p	interpolation: 'do	kleiner: 'p
homotopie: 'do	intervall: 'do	kleinst: 'p
hooke: 'n	invariant: 'p	koeffizient: 'do
horner: 'n	invarianz: 'do	koerper: 'do
huelle: 'do	invers: 'p	koerperaxiome: 'rg
huelLEN: 'do	inverse: 'do	kombination: 'do
hut: 'do	inversion: 'do	komma: 'do
hyper: 'n	invertierbar: 'p	kommutativ: 'p
hypergeometrisch: 'p	involution: 'do	kommutativgesetz: 'rg
ideal: 'do	irgendein: 'art	kommutator: 'do

kommutier: 'am	laenge: 'do	matrix: 'do
kommutierend: 'p	laesst: 'am	matrixwertig: 'p
kompakt: 'p	lage: 'do	matrizen: 'do
kompatibilitaet: 'do	lagrange: 'n	maxima: 'do
komplement: 'do	landau: 'n	maximal: 'p
komplex: 'p	laplace: 'n	maximum: 'do
komplexitaet: 'do	lass: 'am	maxwell: 'n
komplexwertig: 'p	laut: 'since	mayer: 'n
komponente: 'do	lebesgue: 'n	mehrdeutig: 'p
komponente: 'udo	leer: 'p	mehrdimensional: 'p
komponenten: 'do	legendre: 'n	mehrer: 'qt
kompression: 'do	leibniz: 'n	mehrfach: 'qt
konfidenz: 'do	lemma: 'do	mehrmalig: 'qt
kongruent: 'p	lemma: 'rd	menge: 'do
konjugation: 'do	letzt: 'p	messbar: 'p
konjugiert: 'p	levi: 'n	metrik: 'do
konservativ: 'p	lie: 'n	metrisch: 'p
konstant: 'p	liefer: 'am	mindest: 'n
konstante: 'do	liefern: 'follows	minimal: 'p
konstruier: 'aa	liefert: 'follows	minimier: 'am
konstruiert: 'p	lieg: 'am	minimum: 'do
konstruktiv: 'p	limes: 'do	minkowski: 'n
kontinuierlich: 'p	limiten: 'do	minkowsky: 'n
kontrahierend: 'p	linear: 'p	mit: 'since
kontraktion: 'do	linearisierung: 'do	mittel: 'do
konvention: 'udo	linie: 'do	mittelpunkt: 'do
konvergent: 'p	link: 'dir	mittelwertsatz: 'rg
konvergenz: 'do	linksstetig: 'p	modell: 'do
konvergier: 'am	liouville: 'n	moebius: 'n
konvergierend: 'p	lipschitz: 'n	moeglich: 'p
konvex: 'p	loch: 'do	monom: 'do
konzept: 'udo	loecher: 'do	monoton: 'p
koordinate: 'do	loes: 'aa	monotonie: 'do
kovariant: 'p	loes: 'am	multilinear: 'p
kraefte: 'do	loesbar: 'p	multiplikation: 'do
kraft: 'do	loesung: 'do	multiplikator: 'do
kreis: 'do	logarithmus: 'do	multiplizier: 'aa
kreuz: 'do	logik: 'do	multiplizier: 'am
kriterium: 'do	lokal: 'p	nabla: 'do
kriterium: 'udo	lokalkompakt: 'p	nach: 'since
kritisch: 'p	lorentz: 'n	nacheinander: 'do
kuerz: 'aa	mach: 'am	nachrechn: 'aa
kuerzer: 'p	maechtigkeit: 'do	nachweis: 'udo
kuerzere: 'p	majorante: 'do	naechste: 'p
kuerzung: 'do	majorisiert: 'p	naehe: 'do
kuerzungsregel: 'rg	manchmal: 'qt	naeherung: 'do
kugel: 'do	mannigfaltigkeit: 'do	naemlich: 'but
kurve: 'do	mass: 'do	nah: 'p
ladung: 'do	masse: 'do	natuerlich: 'p

neaerungsweise: 'p	orthonormal: 'p	quader: 'do
neben: 'do	paar: 'do	quadrat: 'do
negativ: 'p	paarweise: 'p	quadratisch: 'p
nenner: 'do	pade: 'n	quadrier: 'aa
neutrale: 'p	parabel: 'do	quadrik: 'do
newton: 'n	parallel: 'p	quasi: 'p
nicht: 'pm	parallelepiped: 'do	quellstaerke: 'do
nicht-: 'pm	parallelotop: 'do	quotient: 'do
nie: 'qt	parameter: 'do	radien: 'do
niemals: 'qt	parametrisiert: 'p	radius: 'do
nirgends: 'qt	partial: 'n	raender: 'do
niveau: 'do	partiell: 'p	raeume: 'do
niveauflaeche: 'do	pathologisch: 'p	raeumen: 'do
niveaulinie: 'do	pauli: 'n	raeumlich: 'p
noetig: 'p	periode: 'do	rand: 'do
norm: 'do	periodisch: 'p	randfrei: 'p
normal: 'n	periodizitaet: 'do	rang: 'do
normal: 'p	permanenz: 'do	rational: 'p
normale: 'do	permutation: 'do	raum: 'do
normalerweise: 'p	permutiert: 'p	real: 'p
normalform: 'do	phase: 'do	realisierung: 'do
normiert: 'p	physikalisch: 'p	rechenregeln: 'rd
normierung: 'do	planck: 'n	rechn: 'aa
null: 'do	plateau: 'do	rechnung: 'do
nulldimensional: 'p	poincare: 'n	recht: 'dir
nullen: 'do	pol: 'do	rechteck: 'do
nullstelle: 'do	polar: 'p	rechteckig: 'p
nullteilerfrei: 'p	polarkoordinate: 'do	reduktion: 'do
numerisch: 'p	polynom: 'do	reduzier: 'am
nun: 'but	polynomial: 'p	reell: 'p
oben: 'dir	positiv: 'p	reellwertig: 'p
oberflaeche: 'do	potential: 'do	reflexiv: 'p
obig: 'dir	potenz: 'do	regel: 'do
obsevable: 'do	potenzgesetze: 'rg	regel: 'fd
offen: 'p	prinzip: 'do	regel: 'rd
offenbar: 'p	problem: 'do	regeln: 'rd
omega: 'do	produkt: 'do	regelung: 'udo
operation: 'do	produktformel: 'rg	regulaer: 'p
operator: 'do	projektion: 'do	reibung: 'do
optimal: 'p	projektiv: 'p	reihe: 'do
optimierung: 'do	projiziert: 'p	reihenfolge: 'udo
orbital: 'do	proportional: 'p	rein: 'p
ordne: 'am	proposition: 'do	rekursiv: 'p
ordnung: 'do	proposition: 'rd	relation: 'do
orientierbar: 'p	prozess: 'do	relation: 'fd
orientiert: 'p	punkt: 'do	relativ: 'p
orientierung: 'do	punktiert: 'p	residuen: 'do
ort: 'do	punktweise: 'p	residuum: 'do
orthogonal: 'p	pythagoras: 'n	resonanz: 'do

rest: 'do	semidefinit: 'p	stereographische: 'p
resultat: 'do	senke: 'do	sternfoermig: 'p
resultierend: 'p	senkrecht: 'p	stetig: 'p
reversibel: 'p	separabel: 'p	stets: 'qt
reversibilitaet: 'do	setze: 'suppose	stieltjes: 'n
reversible: 'p	setzen: 'suppose	stoerungen: 'do
richtig: 'p	signatur: 'do	stokes: 'n
richtung: 'do	signifikant: 'p	stom: 'do
riemann: 'n	signum: 'do	streb: 'am
riemannsche: 'n	simplex: 'do	strecke: 'do
riesz: 'n	sind: 'am	streichen: 'udo
ring: 'do	sind: 'follows	strikt: 'p
rotation: 'do	sind: 'if	stueck: 'do
rueckwaerts: 'p	sind: 'since	stueckweise: 'p
rundung: 'do	singulaer: 'p	stufe: 'do
s: 'dird	singularitaet: 'do	substituier: 'aa
saekular: 'p	sinnlos: 'p	substitution: 'do
saemtlich: 'qt	sinnvoll: 'p	subtraktion: 'do
sard: 'n	skalar: 'do	summand: 'do
sattel: 'do	skalar: 'p	summation: 'do
satz: 'rd	smith: 'n	summe: 'do
satz: 'udo	so: 'hence	summer: 'aa
sch: 'nt	sofern: 'if	superposition: 'do
schaetz: 'aa	somit: 'hence	support: 'do
sche: 'nt	sonderfall: 'do	suprema: 'do
scheibe: 'do	sortier: 'aa	supremum: 'do
schema: 'do	sortiert: 'p	surjektiv: 'p
scher: 'nt	soweit: 'if	sylvester: 'n
sches: 'nt	spalte: 'do	symbol: 'do
schliesslich: 'but	spaltenweise: 'p	symbol: 'udo
schmidt: 'n	sparse: 'p	symmetrie: 'do
schnitt: 'do	spatprodukt: 'do	symmetrisch: 'p
schranke: 'do	spektral: 'p	system: 'do
schreib: 'aa	spektrum: 'do	system: 'udo
schreib: 'am	spezialfaelle: 'udo	tabelle: 'do
schreibweise: 'udo	spezialfall: 'udo	tangente: 'do
schritt: 'do	speziell: 'p	tangential: 'p
schrittweite: 'do	spiegelung: 'do	taylor: 'n
schur: 'n	sprung: 'do	technik: 'do
schwarz: 'n	spur: 'do	teil: 'aa
schwingung: 'do	spurformel: 'rg	teil: 'am
sei: 'suppose	stabil: 'p	teil: 'do
sein: 'am	staerke: 'do	teil: 'n
sein: 'art	stammfunktion: 'do	teilbar: 'p
seite: 'do	standard: 'n	teilchen: 'do
seite: 'udo	stationaer: 'p	teilung: 'do
sekante: 'do	steifigkeit: 'do	tensor: 'do
selbstabbildung: 'do	steigung: 'do	tensorfeld: 'do
semi: 'p	stelle: 'do	term: 'do

term: 'udo	umkehrfunktion: 'do	verletzt: 'am
theorie: 'do	umkehrung: 'do	verletzt: 'p
toeplitz: 'n	umlaeuft: 'am	vernachlaessig: 'aa
topf: 'do	umlauf: 'am	verschiebung: 'do
topologie: 'do	umlauf: 'do	verschieden: 'p
topologisch: 'p	umlaufend: 'p	verschwinde: 'am
total: 'p	umordn: 'aa	verschwindend: 'p
traeger: 'do	umrechn: 'aa	verseh: 'aa
traegheit: 'do	un: 'pm	verseh: 'am
transformation: 'do	unabhaengig: 'p	versieh: 'am
transformier: 'aa	unendlich: 'p	vertaeglich: 'p
transformiert: 'p	ungeaendert: 'p	vertausch: 'aa
transitiv: 'p	ungenauigkeit: 'do	vertausch: 'am
translation: 'do	ungerade: 'p	vertauschbar: 'p
transponier: 'aa	ungleichung: 'do	vertauscht: 'p
transponiert: 'p	ungleichung: 'fd	vertauschung: 'do
transposition: 'do	ungleichung: 'udo	vertauschung: 'udo
trenn: 'aa	ungleichungen: 'fd	verteilung: 'do
trennung: 'do	unitaer: 'p	vertraeglich: 'p
treppe: 'do	unmittelbar: 'p	vervollstaendi: 'aa
trigonometrisch: 'p	unschaerfe: 'do	vervollstaendig: 'am
trilinear: 'p	unten: 'dir	vervollstaendigt: 'p
tripel: 'do	unterschiedlich: 'p	verwend: 'aa
trivial: 'proofline	ur: 'n	verwende: 'aa
trivial: 'p	ursprung: 'do	verwendet: 'p
tschebyshev: 'n	urysohn: 'n	verwendung: 'do
tupel: 'do	variabel: 'p	viel: 'qt
ueberdeck: 'am	variable: 'do	vielfach: 'qt
ueberdeckung: 'do	vektor: 'do	vollstaendig: 'p
uebereinstimm: 'am	vektoriell: 'p	volumen: 'do
ueberfluessig: 'p	vektorwertig: 'p	volumina: 'do
ueberfuehr: 'aa	ver: 'pam	vorausgesetzt: 'p
uebergang: 'do	verallgemeinern: 'am	voraussetzung: 'rd
ueberge: 'aa	verallgemeinert: 'p	voraussetzung: 'udo
ueberlagerung: 'do	verbinde: 'am	vorgegeben: 'p
ueberschaetz: 'am	verbindend: 'p	vorig: 'dir
ueberschreite: 'am	verbindung: 'do	vorschrift: 'do
uebrig: 'p	verbot: 'do	vorzeichen: 'do
uebung: 'proofline	verbunden: 'p	wachs: 'am
um: 'pam	vereinfach: 'aa	wachsend: 'p
umfang: 'do	vereinfach: 'am	wachstums: 'do
umfass: 'am	vereinigt: 'p	waechs: 'am
umform: 'aa	vereinigung: 'do	waehl: 'aa
umformulierung: 'udo	verfahren: 'udo	waehl: 'am
umformung: 'do	verfahrens: 'do	wahl: 'fd
umgebung: 'do	vergleich: 'aa	wechsel: 'aa
umgekehrt: 'p	verhalten: 'udo	wechsel: 'do
umkehr: 'do	verkleiner: 'aa	weg: 'do
umkehrbar: 'p	verlangt: 'p	wegen: 'since

wegfaell: 'am	wo: 'if	zusaetzlich: 'p
wegfall: 'am	wohldefiniert: 'p	zusammenfass: 'aa
weggelass: 'aa	wuerfel: 'do	zusammengesetzt: 'p
weggelassen: 'p	wurzel: 'do	zusammenhaengend: 'p
weglass: 'aa	zaehler: 'do	zusammenhang: 'do
wegunabhaengig: 'p	zahl: 'do	zusatz: 'do
weierstrass: 'n	zahlreich: 'qt	zustand: 'do
weil: 'because	zeigt: 'follows	zwang: 'do
welle: 'do	zeile: 'do	zwei: 'qt
weniger: 'p	zeit: 'do	zweideutige: 'p
wenigstens: 'p	zeitabhaengig: 'p	zweidimensional: 'p
wenn: 'if	zentrum: 'do	zweier: 'qt
wert: 'do	zerleg: 'aa	zweierlei: 'qt
werte: 'do	zerleg: 'am	zweifach: 'p
wesentlich: 'p	zerlegung: 'do	zweimal: 'qt
widerspricht: 'am	ziel: 'do	zweit: 'p
widerspruch: 'n	ziffer: 'do	zweite: 'nr
windung: 'do	zu: 'pam	zweiten: 'nr
winkel: 'do	zufallsvariable: 'do	zwischen: 'p
wirbel: 'do	zugehoerige: 'p	zwischenwertsatz: 'rg
wird: 'am	zuordn: 'aa	zyklisch: 'p

## 5 Endings lexicon

-te: 'te	em: 'qd	ender: 'amt	es: 'qd
-ten: 'te	en: 'ad	endes: 'amt	m: 'pp
-tes: 'te	en: 'ap	er: 'ad	n: 'ap
e: 'ad	en: 'op	er: 'op	n: 'op
e: 'ap	en: 'pp	er: 'pp	n: 'pp
e: 'op	en: 'qd	er: 'qd	s: 'op
e: 'pp	end: 'amt	ern: 'op	t: 'ap
e: 'qd	ende: 'amt	es: 'ad	te: 'te
em: 'ad	endem: 'amt	es: 'op	ten: 'te
em: 'pp	enden: 'amt	es: 'pp	tes: 'te

## 6 Phrase lexicon

als uebungsaufgabe: "proofline  
also ist: "hence  
aufgrund von: "since  
aus diesem grund: "hence  
beliebig oft: "qt  
beliebig viel: "qt  
d.h. es ist: "hence  
das eben bewiesene: "rl  
das heisst: "hence  
dem eben bewiesenen: "rl  
den wir spaeter waehlen werden: "prooflineadd

denn es gilt: "since  
denn es ist: "since  
der gesamtausdruck: "rl  
endet hier der prozess: "proofline  
endlich viel: "qt  
ergibt sich: "follows  
ergibt sich aus: "since  
erhaelt man: "follows  
erhalten wir: "follows  
es bleibt zu zeigen: "proofline  
es genuegt zu zeigen: "proofline  
es reicht zu zeigen: "proofline  
fast all: "qt  
finden wir: "follows  
findet man: "follows  
folgt aus: "since  
gleichheit kann nur dann gelten: "proofline  
im dem fall, dass: "if  
im fall: "if  
im fall, dass: "if  
im mittel: "ql  
kann man erreichen: "prooflineadd  
man sieht daraus: "proofline  
man zeigt naemlich induktiv: "proofline  
mehr als ein: "qt  
nach dem eben bewiesenen: "rl  
nehmen wir an: "suppose  
nehmen wir an, dass: "if  
reicht es, zu zeigen: "prooflineadd  
sehen wir: "follows  
setzen wir: "suppose  
sieht man: "follows  
so muessen wir zeigen: "prooflineadd  
unendlich viel: "qt  
vorausgesetzt, dass: "if  
wenn wir zeigen: "prooflineadd  
wir erhalten: "follows  
wir muessen zeigen: "proofline  
wir nehmen an: "suppose  
wir nehmen daher an: "proofline  
wir nutzen aus: "proofline  
wir setzen: "suppose  
wir zeigen: "proofline  
wir zeigen zunaechst: "proofline  
zeigt dass: "follows

## Part III

# The grammar

## 7 Lexical productions

**Productions for "aa** (actions that algorithms (“we”) take)

```
"aa = mit 'aa | 'aa "r | 'aa mit | 'aa von | nach 'aa | 'aa nach  
      | durch 'aa | 'aa des 'o | 'aa mit "r | 'aa und 'v | 'aa von "r  
      | 'aa von 'v | 'aa durch 'f | 'aa durch 'v
```

**Productions for "am** (actions that objects take)

```
"am = 'am | nicht 'am
```

**Productions for "but** (variations of “but”)

```
"but = 'but | 'empty | "but "but
```

**Productions for "dir** (direction)

```
"dir = 'dir'dird
```

**Productions for "f** (formula)

```
"f = 'fd 'f | 'art 'f | 'f (f.ue.) | 'f und 'f | fuer "p "v | 'f sowie 'f  
      | 'f fast ueberall | in der form 'f | 'f fuer "qt "v  
      | 'f fuer ein geeignetes "v
```

**Productions for "follows** (variations of “follows”)

```
"follows = &phrases | 'follows | es 'follows | 'follows 'follows
```

**Productions for "hence** (variations of “hence”)

```
"hence = &phrases | 'empty | 'hence
```

**Productions for "if** (variations of “if”)

```
"if = &phrases | 'if
```

**Productions for "n** (name modifiers)

```
"n = 'n | 'art 'n
```

**Productions for "o** (object, either 'do or 'udo)

"o = 'o | 'o "ql | 'art 'o | 'o von 'o | "o der "p "o

**Productions for "p** (property)

"p = 'p | 'empty | "p "p | "p "ql | nicht 'p | nach "dir 'p

**Productions for "ql** (qualification of 'am and 'p)

"ql = &phrases | 'empty | in "v | um "v | zu "v | auf "v | von "v  
| bzgl. "v | durch "v | ueber "v | bezueglich "v | zugehoerige "o  
| im "o "v | in der "o | der "o "v | im punkt "v | relativ zu "v  
| eingeschraenkt auf "v | mit der "o "v | von "v auf "v  
| von "v nach "v | an der stelle "v | von "v ueber "v  
| zu "v gehoerigen "o | von "v an der stelle "v

**Productions for "qt** (quantification)

"qt = &phrases | "v | 'qt | "v-fach | "v mal | nur 'qt | genau 'qt  
| nicht 'qt | 'qt beliebig | hoechstens 'qt | mindestens 'qt

**Productions for "r** (reference)

"r = "rg | "rl | in "r | zu "r | zum "r | zur "r | 'rd "r | 'art "r  
| "o in "r | "r bis "r | "r von "v | "r fuer "v statt "v

**Productions for "rg** (reference outside the context)

"rg = 'rg | lie-algebra-'rd | "n 'rd | "o 'rd | 'rd von "n | 'rd von "o  
| 'rd fuer "o | 'rd ueber "o | 'rd 'art 'o

**Productions for "rl** (reference to something in the same context)

"rl = &phrases | 'rl | beide 'fd | "dir seite | 'art 'nr 'do  
| 'art 'nr 'rd | 'art "dir 'rd | 'art "dir seite  
| 'art angegebene 'rd | 'art geforderte 'rd | 'art umgekehrte 'fd  
| 'art "o in "r | auf der "dir seite | schritt "v des obigen "o  
| 'art entsprechenden "r fuer "o

**Productions for "since** (variations of "since" or "by")

"since = &phrases | 'since

**Productions for "suppose** (variations of "suppose")

"suppose = &phrases | 'suppose | es 'suppose

### Productions for "v (variable)

```
"v = 'v | 'art 'v | 'v mit "f | 'v fuer 'v
```

## 8 Global productions

### Productions for "sentence (complete sentence)

```
"sentence = "context | "proofline | "statement | "defcomment  
            | "defsentence | "def1left "def1right | "def2left "def2right  
            | "equileft "equiright | "implicleft "implicright  
            | "proofline "prooflineadd | "statement "statementadd
```

### Productions for "context (a sentence that establishes some context)

```
"context = "suppose "v "o.  
"context = "v "suppose "o.  
"context = "v "suppose "p.  
"context = "v "suppose "o "v.  
"context = "v "suppose "o, "f.  
"context = "suppose "v "o, der zu "qt "v auch "v "am.  
"context = es "suppose "f, und "v "suppose eine von 'art "p "o "v  
            abhaengige "o.  
"context = "suppose "v "o, d.h. "o, in dem "qt "o mit "f "o von "v ist.
```

### Productions for "equileft (the first part of an equivalence)

```
"equileft = "o ist genau dann "p,  
"equileft = "v ist genau dann "p,  
"equileft = dann ist "o "f genau dann "qt "v "p,  
"equileft = eine in "v "p "o "v ist genau dann "p,
```

### Productions for "equiright (the second part of an equivalence)

```
"equiright = "if er zu "v "p ist.  
"equiright = "if "v fuer "qt "v "p ist.  
"equiright = "if "o "f nur die "p "o "am.  
"equiright = "if "qt der beiden "p "o "f gilt.  
"equiright = "if es zu "qt "v "o "v gibt mit "f.
```

### Productions for "implicleft (the first part of an implication)

```
"implicleft = "if "f,  
"implicleft = "if "v "o,  
"implicleft = "if "v "am,
```

```

"implicleft = "if "f mit "f,
"implicleft = "if "qt "p "f,
"implicleft = "if "v "p bleibt,
"implicleft = "if "v in "v "p,
"implicleft = "if "v und "v "o,
"implicleft = und "if "v "p ist,
"implicleft = "if "f und hat "v lauter "p "o "v,
"implicleft = "if "v in "o "p und ist "o "v "p,

```

### Productions for "implicright (the second part of an implication)

```

"implicright = auch "f.
"implicright = so "follows "f.
"implicright = "follows stets "f.
"implicright = so "follows ausserdem "f.
"implicright = so "follows auch "f "o.
"implicright = so "follows fuer "v "f.
"implicright = so "follows "f fuer "p "v.
"implicright = so gibt es "qt "o "v mit "f.
"implicright = so "follows dasselbe auch fuer "v und "v.
"implicright = "follows auch "v "p und zwar ist "v "o von "v.
"implicright = so "follows "o "v "p und fuer "o "v "follows "o "f.
"implicright = so gibt es "o "v mit "f derart, dass "v "qt "o hat.
"implicright = so ist "v genau dann "p "if die fuer "f durch "f
    definierten "o "v "am und "p sind.

```

## 9 Definition productions

### Productions for "defcomment (some comment to a definition)

```

"defcomment = man schreibt dann "f.
"defcomment = statt "v schreibt man auch manchmal "v.
"defcomment = wir bezeichnen "o in der regel mit "n.
"defcomment = "if "v "p, so fordert man zusaetzlich "o "v.
"defcomment = "o werden vorzugsweise mit "o "v, "v, 'dots bezeichnet.
"defcomment = sofern mehrdeutigkeit ausgeschlossen ist, ist auch "o "v
    ueblich.
"defcomment = wir schreiben "o in der regel als "v oder einfach "v.

```

### Productions for "defleft (the first part of a definition of type 1)

```

"defleft = "o ist "o,
"defleft = "o heisst "p,
"defleft = "v heisst "o,
"defleft = "v heisst "p,

```

```

"def1left = man schreibt "f
"def1left = "o "v ist "o,
"def1left = "o ist "o "v,
"def1left = "o ist "o "v,
"def1left = "o "v heisst "p,
"def1left = "o "v heisst "p,
"def1left = "qt "o heissen "p,
"def1left = "o mit "f heisst "v,
"def1left = mit "v bezeichnen wir "o,
"def1left = eine "o "v von "o heisst "p
"def1left = "v heisst "o oder "o "o "v,
"def1left = wir schreiben (und sagen) "f ( "v "p "v )
"def1left = wir nennen eine durch "o "v gegebene "o "v in "v "p,

```

### Productions for "def1right" (the second part of a definition of type 1)

```

"def1right = "if "f.
"def1right = "if "f gilt.
"def1right = "if "f und "f.
"def1right = "if "v "p ist.
"def1right = "if "r "p sind.
"def1right = "if er mit "o "am.
"def1right = "if fuer "v "am: "f.
"def1right = in dem "v und "v "am.
"def1right = "if "o "v mit "f "am.
"def1right = "if fuer "o das "r "am.
"def1right = "if "qt "v diese eigenschaft haben.
"def1right = "if es "o "v gibt mit "f.
"def1right = der zu "qt "v auch "v "am
"def1right = so dass die "r bis "r "am.
"def1right = "if "v "o von "v "ql ist.
"def1right = "if das "r gilt, d.h. "f "am.
"def1right = "if fuer "v 'art "dir "o "am: "f
"def1right = "if die durch "f definierte "o "p ist.
"def1right = "if "o "v und "o "v mit "f "am.
"def1right = "if zu "qt "v "o "v "am mit "f.
"def1right = "if es "o "v gibt, so dass "f gilt.
"def1right = "if es "o gibt, die "v auf "v "am.
"def1right = "if es "o "v mit "f gibt, 'art "o heisst "o.
"def1right = in dem "v und zu je zwei "o "v auch "v "am
"def1right = "if sie "p ist, in "o "o "v liegt und dort zur "o "v "p ist.
"def1right = "if man aus "qt "o von "o "v mit "f "qt "o "v so auswaehlen
                kann, dass immer noch "f "am.

```

### **Productions for "def2left** (the first part of a definition of type 2)

```
"def2left = "if "f,  
"def2left = "if "v "o,  
"def2left = "if "v "o und "v "o  
"def2left = "o, das in "qt "o "v von "v mit "f enthalten ist,  
"def2left = 'art "o "v, in dem "v und "v mit "r "p sind, und wo "qt "o "p  
sind,
```

### **Productions for "def2right** (the second part of a definition of type 2)

```
"def2right = heisst "o.  
"def2right = so heisst "o "v "o.  
"def2right = spricht man auch von "o.  
"def2right = wir bezeichnen sie mit "v.  
"def2right = so heisst "o "v mit "f "o.  
"def2right = so wird "since "v "f "o "am.  
"def2right = so heissen "o "o "f "o von "v.
```

### **Productions for "defsentence** (a sentence that defines something)

```
"defsentence = "v heisst "o "o.  
"defsentence = "o "v heissen "o.  
"defsentence = "v heisst dann "o.  
"defsentence = "v heisst dann "v.  
"defsentence = "o ist "o mit "f  
"defsentence = "o "o ist "o "o.  
"defsentence = "o von "v hiessen "o.  
"defsentence = fuer "f heisst "v "o.  
"defsentence = "v heisst "p, "if "f.  
"defsentence = "qt solche "o heisst "o.  
"defsentence = wir schreiben "f falls "f.  
"defsentence = "v bezeichnet "o aller "v.  
"defsentence = "o sind "o der form "f.  
"defsentence = man nennt "r "o von "v.  
"defsentence = "if "f heisst "v ein "o.  
"defsentence = "o der form "f heisst "o.  
"defsentence = "v heisst "o 'art "o "v.  
"defsentence = ein "o heisst "p, "if "f.  
"defsentence = statt "v schreibt man auch "v.  
"defsentence = "o wird kurz als "o bezeichnet.  
"defsentence = "qt "o mit "o "v ist "o.  
"defsentence = "f heisst "v von "v und "v.  
"defsentence = mit "v bezeichnen wir "o "o "f.  
"defsentence = wir schreiben "f, "if "v "p ist.  
"defsentence = die durch "r definierte "o heisst "o.  
"defsentence = ein "o ist "o "v "o mit "f.
```

```

"defsentence = aus "r folgt "f, d.h. "v ist "p.
"defsentence = das "p "o ist "p "p "since "f.
"defsentence = den ausdruck "v in "r nennt man "o.
"defsentence = statt "f sagt man auch: "v wird "p.
"defsentence = "o "v wird als "o von "o bezeichnet.
"defsentence = "f ist eine abkuerzung fuer "v ist "p.
"defsentence = "qt "o mit der eigenschaft "r heisst "o.
"defsentence = "qt "o mit der eigenschaft "r heisst "p.
"defsentence = die durch "r definierte "o "v heisst "o.
"defsentence = "v heisst dann "o, und "v "o von "v.
"defsentence = hat "qt "o "qt "o, so heisst "v "p.
"defsentence = "qt "o heisst "p, "if es "ql "o "am.
"defsentence = "o der "v "f ist "p durch "f, "f "p.
"defsentence = 'art "v ist fuer "f durch die "v "f "p.
"defsentence = in diesem fall sagen wir, "v und "v sind "p.
"defsentence = die sich fuer "v ergebende "o heisst "o "o "v.
"defsentence = "qt "o mit der in "r geforderten eigenschaft heisst "o.
"defsentence = fuer "f definieren wir "o "f und schreiben abkuerzend "f.
"defsentence = man sagt, zwei "o "v und "v sind "p, "if "f.
"defsentence = man schreibt statt "f auch "f und nennt "v das "v.
"defsentence = "o "qt "o "ql heisst "o, und wird mit "v bezeichnet.
"defsentence = "qt "o ist ein durch "f definiertes "o, dabei ist "f.
"defsentence = "o auf "o "v ist ein "o "v von "o mit "f.
"defsentence = "o "o "v ist "p als "o des durch "r "p "o.
"defsentence = es gibt "qt "o mit "p "o "v, naemlich "o mit "f.
"defsentence = fuer "v bezeichnen wir mit "v die durch "f definierte "o "v.
"defsentence = "o ist "o "v mit "o "v wobei "v "o "o "v ist.
"defsentence = "o "p "o 'dash kurz "o 'dash ist "o mit "o im "o "v.
"defsentence = "o "v bezeichnet "o, die "v und "v "am und die "p "o "am.
"defsentence = dann bezeichnet "v die aus "o "v durch "o "o mit "o "v
    entstehende "o.
"defsentence = "v bezeichne "o, die aus "v durch "o der "p "o und der "p
    "o "am.
"defsentence = man sagt dann, "v "am den "o "v und nennt "o "o "v mit
    dieser eigenschaft "o von "v.

```

## 10 Statement productions

### Productions for "statement (statement)

```

"statement = es gilt "f
"statement = es gilt: "f
"statement = "f "since "f
"statement = insbesondere ist "f
"statement = "f, so ist "f
"statement = "qt "o ist "p
"statement = es gibt "qt "v

```

```

"statement = es gilt stets "f
"statement = fuer "o "v "follows
"statement = fuer "v "follows "f
"statement = "v ist "p und "p
"statement = "o "v "am aus "v
"statement = "qt "o "v ist "p
"statement = "v "am "o mit "f
"statement = "v und "v sind "o
"statement = "qt "o hat "qt "o
"statement = fuer "f ist auch "f
"statement = in diesem fall ist "f
"statement = in diesem fall gilt "f
"statement = "v "suppose "o von "v
"statement = in "qt "o "follows "f
"statement = "p "o haben dasselbe "o
"statement = die "dir aussagen sind "p
"statement = fuer "v "follows "qt "f
"statement = ausserdem gilt fuer "f: "f
"statement = insbesondere ist "o "v "o
"statement = "aa "o durch ihr "v-faches
"statement = ausserdem "follows "since "r "f
"statement = "o "am "o auf "o ab
"statement = "o ist stets im "o "p
"statement = "qt "o "f ist ein "o
"statement = "o "v 'art "o ist "o
"statement = "qt "o hat "qt "o "ql
"statement = "o := "f erfuehlt die "r
"statement = "o "v ist "since "r "p
"statement = nicht "qt "o hat "qt "o
"statement = 'art "o "am "qt "o "v
"statement = "qt "p "o besitzt "qt "o
"statement = "since "qt "v "am "f "o
"statement = in "qt "o "follows die "r
"statement = fuer "v "follows die "r "f
"statement = in dem fall "follows "qt "o
"statement = man schreibt in diesem fall "f
"statement = d.h. "o stimmt mit "o ueberein
"statement = dann "follows fuer "v "o: "f
"statement = "qt "o ist von der form "r
"statement = es gibt "qt "o "ql mit "f
"statement = "qt "o hat "qt der formen "v
"statement = fuer "qt "v gelten "n aus "r
"statement = "since "f "am "o "v "o, "o
"statement = hat "v "o "v, so "follows "f
"statement = dies schliesst "o "f und "f ein
"statement = die dadurch definierte "o "v ist "p
"statement = insbesondere ist fuer "f "qt "o "p
"statement = dann "follows fuer "v "dir 'rd: "f
"statement = insbesondere "follows "v fuer "o "v "p

```

```

"statement = insbesondere "follows mit "v auch "v "p
"statement = fuer "p "v hat "f "qt "o "f
"statement = "o ist in diesem fall "o "o "v
"statement = "o von "v sind genau "o von "v
"statement = in "qt "o gelten fuer "o die "r
"statement = "o "qt "o braucht nicht "p zu sein
"statement = 'art "p "o ist "since 'art "o "p
"statement = "since "v "follows dann "f mit "o "v
"statement = "if "qt "p "f so "follows ausserdem "f
"statement = "o 'art "o "f in "o "f ist "p
"statement = "v ist mit 'art "p "o ein "p "o
"statement = "qt "o "ql "am sich nur um ein "o
"statement = "since "f hat "o mit "o "f "o "f
"statement = fuer "f "follows "qt "o "v und "v "p
"statement = fuer "qt "o "v hat "o "f denselben "o
"statement = es gilt "f wobei "o mit "f wegzulassen sind
"statement = "qt "v hat "qt "o der form "f mit "f
"statement = "since "f wird 'art "p "o "v auf "v "am
"statement = "qt "o laesst sich "p in der form "f schreiben
"statement = "qt "o "ql laesst sich "p als "o "f schreiben
"statement = es gibt "qt "o "f mit "o "v und "o "v
"statement = in "qt "o "v gibt es "qt "o "v mit "f
"statement = "qt "o "v laesst sich "ql "p in "o "am, "f
"statement = "qt "o "v laesst sich als "o von "qt "o schreiben
"statement = "qt "o in 'art "v "o nimmt dort "o und "o an
"statement = "qt "p "o "v kann "p durch "p "o "v "p werden
"statement = aus einer "o "f bekommt man "v "p "since "aa "o "v
"statement = es gilt "f, wobei "v als "o mit "o "v aufgefasst wird
"statement = "if "f und "if "v in "v "p, so "follows ausserdem "f
"statement = es gibt "qt "o "v "ql mit vorgegebenen "o, naemlich "o mit "f
"statement = zu je "v "p "p "o "v gibt es "qt "o "v mit "f
"statement = dann ergibt sich 'art "p "o "v der "f "p "aa der "p "f
"statement = "since "qt "o "v ist die durch "f definierte "o eine "o "ql "p
"statement = "o eines "o "v ist "o aller "v, fuer die es "v "o in "v gibt
"statement = "since "qt "o "o "f mit "o "v und "o "v nimmt "o "f "p ab

```

### **Productions for "statementadd** (some additional information to a statement)

```

"statementadd = "f
"statementadd = 'punct
"statementadd = und "f
"statementadd = und "r
"statementadd = "hence "f
"statementadd = und heisst "o
"statementadd = und "f "if "f
"statementadd = und fuer "f "f
"statementadd = hat "hence "o "v
"statementadd = und es "follows "f

```

```

"statementadd = und es "follows "r
"statementadd = insbesondere ist "o "p
"statementadd = der mit "v bezeichnet wird
"statementadd = und wird mit "v bezeichnet
"statementadd = so "aa sich "v mit "v
"statementadd = naemlich die durch "f "p "o
"statementadd = 'art "o braucht "but nicht "p sein
"statementadd = insbesondere ist "o, "if es "am, "p
"statementadd = und fuer "f laesst sich "v "p aus "f "am
"statementadd = diese ist "p von der "p "o von "v als "o "p
"statementadd = und laesst sich durch hinzunahme geeigneter "o zu einer "o
    von "v "aa
"statementadd = so dass "o "r fuer "qt "v mit "f "am und fuer "qt "v mit
    "f "am
"statementadd = indem man die "v "o "v mit "f "p "am, und die uebrigen "o
    "p aus "f "am
"statementadd = indem man fuer "v von der "p "o von "v das "v-fache "o "v
    aller zeilen von "v abzieht

```

## 11 Proof productions

**Productions for "proofline** (a sentence in a proof)

```

"proofline = &phrases
"proofline = 'proofline
"proofline = "r gilt
"proofline = fall "r
"proofline = "but "f
"proofline = waere "f
"proofline = "since "f
"proofline = "r "follows
"proofline = "suppose "f
"proofline = "suppose "v
"proofline = es ist "f
"proofline = "r ist "p
"proofline = "v hat "o
"proofline = "v ist "p
"proofline = ist "v "o
"proofline = ist "v "p
"proofline = nun ist "f
"proofline = "r ist klar
"proofline = hier ist "f
"proofline = "aa nach "v
"proofline = analog zu "r
"proofline = ist "but "f
"proofline = waere "f "p
"proofline = waere "v "o

```

"proofline = "r besagt "f  
 "proofline = bleibt "v "p  
 "proofline = haette "v "o  
 "proofline = weiter ist "f  
 "proofline = "aa ergibt "f  
 "proofline = die 'rd gelten  
 "proofline = "hence "am "r  
 "proofline = "r gehen analog  
 "proofline = analog folgt "f  
 "proofline = wir benutzen "r  
 "proofline = "f "follows "r  
 "proofline = "suppose "v "p  
 "proofline = "v "suppose "p  
 "proofline = schreiben wir "f  
 "proofline = "aa "follows "r  
 "proofline = "r "follows ebenso  
 "proofline = schliesslich ist "f  
 "proofline = tatsaechlich ist "f  
 "proofline = "hence "follows "r  
 "proofline = insbesondere gilt "f  
 "proofline = schreiben wir "hence  
 "proofline = daraus "follows "but  
 "proofline = o.b.d.a. "suppose "f  
 "proofline = offensichtlich ist "f  
 "proofline = "suppose zunaechst "f  
 "proofline = offensichtlich gilt "r  
 "proofline = zunaechst "suppose "f  
 "proofline = "r "follows unmittelbar  
 "proofline = "aa wir "o um  
 "proofline = ist "v mit "f  
 "proofline = um "r zu zeigen  
 "proofline = "but "v ist "p  
 "proofline = ist "v 'art "o  
 "proofline = dieses "o ist "p  
 "proofline = "hence "o ist "p  
 "proofline = "hence ist "r "p  
 "proofline = "hence ist "v "p  
 "proofline = "o ist "hence "f  
 "proofline = "since "f ist "f  
 "proofline = fuer diese gilt "f  
 "proofline = "hence "am "r "p  
 "proofline = "r gilt "since "r  
 "proofline = "since "f wird "f  
 "proofline = in "r wurde benutzt  
 "proofline = "hence muss "f "am  
 "proofline = genauso sieht man "f  
 "proofline = ist umgekehrt "v "o  
 "proofline = wie "r "suppose "f  
 "proofline = 'because "v "p ist

"proofline = 'because "v "p war  
 "proofline = "if "f "follows "f  
 "proofline = analog ergibt sich "r  
 "proofline = fuer "f "follows "r  
 "proofline = fuer "f "suppose "f  
 "proofline = gilt gleichheit in "r  
 "proofline = sind umgekehrt "v "p  
 "proofline = fuer "r folgt dasselbe  
 "proofline = "r wird analog bewiesen  
 "proofline = "r "follows "since "f  
 "proofline = "since "f "follows "f  
 "proofline = "since "r "follows "p  
 "proofline = ist umgekehrt "r richtig  
 "proofline = "hence "follows auch "f  
 "proofline = offensichtlich ist "v "p  
 "proofline = umgekehrt "follows aus "f  
 "proofline = "since konstruktion ist "f  
 "proofline = beides zusammen "follows "f  
 "proofline = "f ist "p zu "f  
 "proofline = "o "o ist "p "p  
 "proofline = "o "f "am nun "f  
 "proofline = "o "v "am "r "r  
 "proofline = "o "v "am aus "o  
 "proofline = "p ist "qt "o "o  
 "proofline = "r ist "p fuer "f  
 "proofline = "r und "r sind "p  
 "proofline = fuer "f ist "o "p  
 "proofline = fuer "f ist "r "p  
 "proofline = fuer "f ist "v "p  
 "proofline = ist "f und etwa "f  
 "proofline = dass das "p "o gilt  
 "proofline = fuer "f "aa wir "n  
 "proofline = ist "v eine 'nr "o  
 "proofline = wenden wir "r "v an  
 "proofline = dabei "am "o aus "v  
 "proofline = gilt "r fuer "qt "v  
 "proofline = wie im "r folgt "but  
 "proofline = "since "f ist "r "p  
 "proofline = "since "f ist "v "p  
 "proofline = "since "r ist "r "n  
 "proofline = "aa und "aa ergibt "f  
 "proofline = "hence "am "r und "r  
 "proofline = das folgt sofort aus "r  
 "proofline = im "o "f "follows "f  
 "proofline = "hence ist "v schon "p  
 "proofline = "since "f gilt auch "r  
 "proofline = "since "r "am auch "f  
 "proofline = "since "r "am sich "f  
 "proofline = "v "suppose "o und "f

"proofline = "v bleiben wegen "f "p  
 "proofline = nun "follows "r aus "r  
 "proofline = "since "f ist alles klar  
 "proofline = aus "r, angewandt mit "f  
 "proofline = bezeichnen wir "o mit "v  
 "proofline = passiert dies fuer ein "v  
 "proofline = wir bezeichnen mit "v "o  
 "proofline = "hence ist "f "since "r  
 "proofline = dann betrachten wir "o "v  
 "proofline = fuer "v "follows dann "f  
 "proofline = wieder "r gilt "hence "f  
 "proofline = 'because "v 'art "o ist  
 "proofline = "follows sofort "aa in "r  
 "proofline = "r "follows direkt aus "r  
 "proofline = "since "f gilt "hence "f  
 "proofline = "since "r direkt "r folgt  
 "proofline = andernfalls gibt es "qt "v  
 "proofline = es reicht "hence, zu zeigen  
 "proofline = fuer "f passiert gar nichts  
 "proofline = wir definieren "v durch "r  
 "proofline = "follows in "qt faellen "f  
 "proofline = "since "f "follows "o "o  
 "proofline = "since "o berechnen wir "f  
 "proofline = fuer festes "v "suppose "f  
 "proofline = fuer festes "v "suppose "f  
 "proofline = "hence "follows "r fuer "f  
 "proofline = "since "f "follows dann "f  
 "proofline = "since "r erfordert dies "f  
 "proofline = schliesslich "am "r "aa "r  
 "proofline = zusammen mit "r "follows "f  
 "proofline = "since "f "follows "but "f  
 "proofline = "since "r "follows "but "f  
 "proofline = beidesmal "follows "o aus "r  
 "proofline = sind 'art "v "qt verschieden  
 "proofline = 'because "o "o nicht aendern  
 "proofline = "hence koennen wir "r benutzen  
 "proofline = "o "follows daraus "since "o  
 "proofline = "f beweisen wir durch widerspruch  
 "proofline = indem man gegebenenfalls "o hinzunimmt  
 "proofline = es ist "f mit "o "f  
 "proofline = "r ist "o "f von "r  
 "proofline = "r ist nur "o von "r  
 "proofline = nun ist "v von "v "p  
 "proofline = "aa man "r "v mit "v  
 "proofline = setzt man "v in "f ein  
 "proofline = sind "o "v von "v "p  
 "proofline = ist "v 'art "p "o "v  
 "proofline = setzen wir "r in "r ein  
 "proofline = fuer "v aus einer "p "o

"proofline = waehlen wir nun "v so "p  
 "proofline = "aa wir hier "v durch "v  
 "proofline = "hence ist "v "p und "p  
 "proofline = "since "f hat "o "o "v  
 "proofline = fuer "f "am "o gegen "v  
 "proofline = fuer "p "v findet man "f  
 "proofline = wir waehlen "o "v von "v  
 "proofline = "f und "r beweist man "aa  
 "proofline = "o einer "o ist wieder "p  
 "proofline = "since "f ist "f oder "f  
 "proofline = auf "o findet man "but "o  
 "proofline = wir erhalten "o "v mit "f  
 "proofline = "r "am "v "hence "qt "o  
 "proofline = "since "r gilt "qt der "f  
 "proofline = "since "r sind "qt "o "p  
 "proofline = durch "v wird 'art "o "am  
 "proofline = treten dabei "o "v "qt auf  
 "proofline = wir wenden hier direkt "r an  
 "proofline = "hence "am "v 'art "p "o  
 "proofline = "hence kann "o fuer "f "am  
 "proofline = haben "v und "v dasselbe "o  
 "proofline = in "qt "o "follows fuer "v  
 "proofline = wir betrachten "o "v mit "f  
 "proofline = "f ist als axiom "r gefordert  
 "proofline = mit "v statt "v "follows "f  
 "proofline = "hence erfuehlt "v die "r "f  
 "proofline = "since "f koennen wir "r "aa  
 "proofline = fuer "v koennen wir "f waehlen  
 "proofline = schliesslich ist "v "o von "f  
 "proofline = 'because wir "r in "o rechnen  
 "proofline = "aa der beiden seiten ergibt "f  
 "proofline = analog wie "r kann man erreichen  
 "proofline = wir "aa zunaechst "o "v derart  
 "proofline = wir schreiben "hence "r als "f  
 "proofline = "aa von "v koennen wir erreichen  
 "proofline = "hence "am "v keine anderen "o  
 "proofline = andernfalls suchen wir uns "o "v  
 "proofline = angenommen, "r gilt fuer "qt "v  
 "proofline = in "o lautet "f ausfuehrlich "f  
 "proofline = "hence hat "v die behaupteten "o  
 "proofline = "since "r "follows dann auch "f  
 "proofline = diese aussage gilt "but genau dann  
 "proofline = "since "f muss "o gewaehlt werden  
 "proofline = "r folgt "since widerspruch aus "r  
 "proofline = "since "f "follows "f "since "r  
 "proofline = angenommen, die behauptung gilt fuer "f  
 "proofline = ohne beschraenkung der allgemeinheit "suppose "v  
 "proofline = "aa von "o und "o der "f  
 "proofline = "o "v mit "f "am "o "v

"proofline = dazu gibt es "o "v mit "f  
 "proofline = "o von "v hat "but "o "v  
 "proofline = "r ist nur fuer "f der fall  
 "proofline = "r ist klar fuer "f oder "f  
 "proofline = "since "f ist "f "p zu "f  
 "proofline = "since "f ist "r "p zu "f  
 "proofline = "v, "p auf "o, ist dann "p  
 "proofline = auf 'art "o "am "o von "v  
 "proofline = dies zeigt "o von "r und "r  
 "proofline = fuer "f ist mit "v sicher "f  
 "proofline = "aa und neu 'aa ergibt nun "f  
 "proofline = "but "v kann auch "qt "o sein  
 "proofline = "hence "am "r und "r fuer "v  
 "proofline = "hence "am sich "o "r auf "v  
 "proofline = "o "r gelten sogar fuer "p "o  
 "proofline = die "p "o von "v "follows "f  
 "proofline = fuer "v ist "hence "f "if "f  
 "proofline = "aa "o "v fuer festes "v zeigt  
 "proofline = "r ist "p zu der darstellung "f  
 "proofline = "r liegt einer der faelle "f vor  
 "proofline = dazu waehlen wir 'art "v mit "f  
 "proofline = ist "v irgendeine "o mit "o "r  
 "proofline = "hence gibt es "since "r ein "v  
 "proofline = "r ergibt sich "since "f aus "f  
 "proofline = "since "f ergibt sich "r aus "r  
 "proofline = "since "o koennen wir "v so "aa  
 "proofline = ist "v 'art in "v enthaltene "o  
 "proofline = bezeichnen wir 'art "p "v mit "v  
 "proofline = wenn wir nun auch "f zeigen koennen  
 "proofline = wir bestimmen 'art "o "v 'art "o  
 "proofline = zur berechnung "o setzen wir "v ein  
 "proofline = "hence sind "n innerhalb von "v "p  
 "proofline = "since "f "follows "r "aa von "v  
 "proofline = die durch "f definierten "o "am "f  
 "proofline = "since "f fuer "qt "v "follows "r  
 "proofline = "since "r reicht es "hence zu zeigen  
 "proofline = dieser beweis wird "p mit "n gefuehrt  
 "proofline = ein solcher "o "am "since "r sicher  
 "proofline = fuer "v wird durch "f "o beschrieben  
 "proofline = zum nachweis der "p genuegt es "hence  
 "proofline = 'because in "r "o 'art "o auftritt  
 "proofline = "since "r genuegt es "hence zu zeigen  
 "proofline = "follows "aa und "aa auf beiden seiten  
 "proofline = "r "follows "since "r "p "since "r  
 "proofline = wir fuehren den beweis in "v teilschritten  
 "proofline = d.h. "o der geforderten art "am tatsaechlich  
 "proofline = die meissten argumente entsprechen denen in "r  
 "proofline = "v ist die "p "o des "o "v  
 "proofline = als "o von "o ist "v "p "p

"proofline = zu "qt "v gibt es "r "o "v  
 "proofline = "qt "p und "p "p "o ist "p  
 "proofline = fuer "p "v ist "v "o von "f  
 "proofline = ist "v in 'art "p "o "v "p  
 "proofline = ist "v 'art "o in "v und "f  
 "proofline = bilden wir nun "o "v mit "o "v  
 "proofline = durch "r ist "o mit "o "f "am  
 "proofline = "aa in "r ergibt 'aa mit "v "f  
 "proofline = "v "am "but aus "qt "v mit "f  
 "proofline = "hence hat "qt "o in "v 'art "o  
 "proofline = es muss "hence "qt "v mit "f "am  
 "proofline = es reicht also, "v so "p zu waehlen  
 "proofline = "aa man "r durch "v und benutzt "r  
 "proofline = "since "r fuer "v statt "v ist "f  
 "proofline = betrachten wir nur die "p "o als "p  
 "proofline = es reicht, den fall des "o zu beweisen  
 "proofline = "aa man beide "r und "aa das ergebnis  
 "proofline = "aa mit "v wird diese gleichung zu "f  
 "proofline = "r "follows fuer "f und "v statt "v  
 "proofline = fuer die spezielle wahl "v wird "o "n  
 "proofline = setzen wir speziell den wert "v "f ein  
 "proofline = umgekehrt sind die durch "r "p "v "p  
 "proofline = "aa von "f mit "o "follows sofort "f  
 "proofline = da wir schon wissen, wie "o aussehen muss  
 "proofline = 'because fuer "v auch "o "o noetig ist  
 "proofline = fuer "f koennen wir "r in "o ausdruecken  
 "proofline = "since "f ist "but im widerspruch dazu "f  
 "proofline = das "r wird nach "aa abgeschaezt durch "f  
 "proofline = die faelle "f und "f sind dabei eingeschlossen  
 "proofline = man kann "o "v ebenso nach hinten herausziehen  
 "proofline = "since 'art "p "o kann man o.b.d.a. erreichen  
 "proofline = unter allen moeglichen "o waehlen wir eine solche  
 "proofline = "r gibt es "o "v und ein "o "v  
 "proofline = fuer "f ist "v die "p "o mit "f  
 "proofline = nun "aa wir, dass "r "v eine "o "am  
 "proofline = "if "f schreibt sich "r in "o als "f  
 "proofline = "since "f heben sich "o in "r "p weg  
 "proofline = "suppose "v eine "p "o von "v mit "r  
 "proofline = setzen wir "r in "r ein und benutzen "f  
 "proofline = 'because "v und "v "p mit "o "v sind  
 "proofline = "r "follows fuer "f "r, und "p "p "r  
 "proofline = "since "f kann man "o "v durch "v "aa  
 "proofline = "since "r erhaelt man "o auch "p zu "f  
 "proofline = dass "v "p ist, sieht man genauso wie "r  
 "proofline = man kann "v als "o derselben "o "v "aa  
 "proofline = "since "v "o von "v war, "follows "v "o  
 "proofline = "hence muss "v "p und natuerlich auch "p sein  
 "proofline = "since "r "follows ebenso "f und fuer "f "f  
 "proofline = 'because "since konstruktion der "v fuer "f "f gilt

"proofline = indem man gegebenenfalls weitere "o mit koeffizienten null dazunimmt

"proofline = "r "o ist "f und "f "qt "v mit "f

"proofline = an "qt "o kann man "v "p "o "ql "aa

"proofline = wir "aa "v "p "p als "o "qt "o "ql

"proofline = dass "n von "qt "o wieder "o ist, ist "p

"proofline = fuer "o mit nur "qt "o ist die aussage "p

"proofline = "aa "f "follows als "o von "v den "o "f

"proofline = "o "v "am dann die "v, "but nicht "o "v

"proofline = die "v bilden 'art "p "o 'art "p "o "v

"proofline = zum beweis "o gehen wir wieder von "o "r aus

"proofline = "aa "o "v bzw., "v fuer festes "v zeigt "r

"proofline = "aa kuerzt sich "qt "p "o gegen "qt "p weg

"proofline = nehmen wir an, "r gilt schon fuer "v statt "v

"proofline = denn "v und "v sind "since "r nicht beide "v

"proofline = fuer 'art "p "o erhalten wir 'art "p "o "f

"proofline = um "p zu zeigen, schaeetzen wir "v nach unten ab

"proofline = "aa wir nun "p der reihe nach die "p komponente

"proofline = 'because "v und daher "v "o ist, muss "f "am

"proofline = "hence gilt "r fuer "qt "p an "v liegenden "v

"proofline = angenommen, sie gilt fuer "p "o von "v statt "v

"proofline = "since der "p "o ist "since "r "p "am erlaubt

"proofline = je nachdem, ob "f, erhaelt man einen der "v faelle

"proofline = 'because "since wahl "o "o "f fuer "qt "v gilt

"proofline = 'because sich "r beim "aa von "o offenbar nicht aendert

"proofline = "since "f und "r bekommt man naemlich den widerspruch "f

"proofline = "r der "p gibt es also "qt "o "v von "v

"proofline = mit "o "v von "v "am "o "v gegen "v "p

"proofline = "v und "v bilden "hence "p "o auf "p "o ab

"proofline = fuellt man einen "o "v mit nullen zu "o "v auf

"proofline = fuer ein weiters "o "v gelte "f "r und "f "r

"proofline = wir "aa "o "r in "o ueber die einzelnen "o "o

"proofline = "hence ist "f und "r muessen "qt "v mit "f "am

"proofline = umgekehrt ist "o "v mit "f "since "r "p mit "o

"proofline = "o mit "f kann es "but "since "r auch nicht geben

"proofline = "since "o von "v laesst sich das weiter umformen zu "f

"proofline = zum nachweis der "p von "v schaeetzen wir zunaechst "v ab

"proofline = "v laesst sich "hence "since "f als "o der "v schreiben

"proofline = von "o fuer "o muessen "hence nur noch "r nachgeprueft werden

"proofline = 'because "o, ueber "o genommen wird, mit wachsendem "v "p wird

"proofline = wiederholung desselben arguments liefert dasselbe mit "v fuer "f statt "v

"proofline = die "p "o "v hat dann in der "p "o lauter "n

"proofline = als "o zwischen "o "p "o sind "v und "v also "p

"proofline = die "p "o 'art "p "o erhaelt man aus 'art "p "o

"proofline = insbesondere ist "o von "v 'art "p "o 'art "p "o "v

"proofline = alles "follows sofort "r "aa der werte an einer beliebigen stelle "v

"proofline = fuer "p "v "am "qt solche "o in "o von "p "o "v

"proofline = bezeichnen wir mit "v und "v den "p und "p "o von "v  
"proofline = zu je zwei "o "v gibt es "p "o "v und "v mit "f  
"proofline = und "since "f sind "o "o als "o "o "v bis auf "o "p  
"proofline = fuer "qt "o von "v gibt es "since "p "qt "o "v mit "f  
"proofline = fuer "f braucht man zur "o von "v nur die schon "p "v mit "f  
"proofline = nun "suppose "v "o von "v als bruch und "v 'art "p "o von "f  
"proofline = 'because die "dir seite von "r sich bei "o von "v und "v  
nicht aendert  
"proofline = "hence laesst sich "f "since "r in 'art "p "o "ql "p nach "v  
aufloesen  
"proofline = wir ziehen nun von den "o "v von "v das "v-fache von "v ab,  
wo "f  
"proofline = fuer ein "p "o "v bildet die zu "v "p "o "v "o "v in sich ab  
"proofline = "hence kann man "r durch "p "o auf die "p form "f mit "v und  
"v bringen  
"proofline = 'because "v "p ist, ist fuer "qt "v in 'art "p, in "v  
liegenden "o von "v "o "p  
"proofline = "r kann man "v in der form "v mit "o "v von "f und "f und  
einer "o "v mit "n und "n auf "o schreiben

**Productions for "prooflineadd** (some additional information to a sentence in a proof)

"prooflineadd = &phrases  
"prooflineadd = "f  
"prooflineadd = 'punct  
"prooflineadd = "follows  
"prooflineadd = "follows  
"prooflineadd = 'prooflineadd  
"prooflineadd = ist "f  
"prooflineadd = mit "f  
"prooflineadd = und "f  
"prooflineadd = "if "f  
"prooflineadd = dass "f  
"prooflineadd = "hence "f  
"prooflineadd = "hence "f  
"prooflineadd = "hence "o  
"prooflineadd = "hence "o  
"prooflineadd = "since "f  
"prooflineadd = "v sei "o  
"prooflineadd = ist "r "p  
"prooflineadd = ist "v "o  
"prooflineadd = so dass "f  
"prooflineadd = "if "f ist  
"prooflineadd = dass "f ist  
"prooflineadd = so waere "f  
"prooflineadd = dass "f wird  
"prooflineadd = dass "r "am

"prooflineadd = und damit "o  
 "prooflineadd = "hence zu "v  
 "prooflineadd = "hence "v "p  
 "prooflineadd = "r "hence "f  
 "prooflineadd = also sicher "f  
 "prooflineadd = ist "hence "p  
 "prooflineadd = und "hence "f  
 "prooflineadd = und "since "r  
 "prooflineadd = "hence "am "o  
 "prooflineadd = "hence "o "ql  
 "prooflineadd = "hence auch "f  
 "prooflineadd = "hence auch "p  
 "prooflineadd = so "follows "r  
 "prooflineadd = und "follows "f  
 "prooflineadd = und "follows "f  
 "prooflineadd = und "follows "r  
 "prooflineadd = und "r "follows  
 "prooflineadd = "hence selbst "p  
 "prooflineadd = woraus "r "follows  
 "prooflineadd = "hence "follows "f  
 "prooflineadd = "hence "follows "r  
 "prooflineadd = insbesondere "follows "r  
 "prooflineadd = so ist auch "f  
 "prooflineadd = und "v ist "p  
 "prooflineadd = und "v sei "o  
 "prooflineadd = "if "f und "f  
 "prooflineadd = "if "v "p ist  
 "prooflineadd = dass "o "p ist  
 "prooflineadd = dass "v "p ist  
 "prooflineadd = mit der wahl "f  
 "prooflineadd = so waere "f "r  
 "prooflineadd = und "f "if "f  
 "prooflineadd = und "qt "p ist  
 "prooflineadd = "but mit "p "o  
 "prooflineadd = dass "o "f wird  
 "prooflineadd = dass auch "f ist  
 "prooflineadd = "hence es ist "f  
 "prooflineadd = "if man "f setzt  
 "prooflineadd = dass "qt "v "am  
 "prooflineadd = der zu "v gehoert  
 "prooflineadd = und sind daher "p  
 "prooflineadd = "am "hence in "v  
 "prooflineadd = "hence "v ist "o  
 "prooflineadd = "hence "v ist "p  
 "prooflineadd = "hence ist "v "p  
 "prooflineadd = ist "hence "r "p  
 "prooflineadd = und "r "since "f  
 "prooflineadd = und zeigt, dass "f  
 "prooflineadd = enthaelt "v "o "v

"prooflineadd = so "follows "o "f  
 "prooflineadd = so "follows "r "f  
 "prooflineadd = und "hence auch "f  
 "prooflineadd = und erhaelt dann "f  
 "prooflineadd = und es "follows "f  
 "prooflineadd = "am "hence "qt "o  
 "prooflineadd = so dass "r "follows  
 "prooflineadd = und "follows "o "v  
 "prooflineadd = 'because "o "o ist  
 "prooflineadd = 'because "o "p ist  
 "prooflineadd = 'because "v "p ist  
 "prooflineadd = "follows "f fuer "f  
 "prooflineadd = "follows "f fuer "v  
 "prooflineadd = so "follows ("r) "f  
 "prooflineadd = und "aa "follows "r  
 "prooflineadd = 'because "f "r "am  
 "prooflineadd = "hence "f "since "r  
 "prooflineadd = so "follows ebenso "f  
 "prooflineadd = so "follows wieder "f  
 "prooflineadd = 'because sich "o "am  
 "prooflineadd = dass "o angenommen wird  
 "prooflineadd = und daraus "follows "r  
 "prooflineadd = und ebenso "follows "f  
 "prooflineadd = und zeigen zunaechst "r  
 "prooflineadd = "follows "since "r "f  
 "prooflineadd = "hence "follows "r "r  
 "prooflineadd = "since "r "follows "f  
 "prooflineadd = insbesondere sind "v "p  
 "prooflineadd = und daher "p voneinander  
 "prooflineadd = "follows dasselbe fuer "r  
 "prooflineadd = dass "r tatsaechlich "follows  
 "prooflineadd = so hat "v "o "v  
 "prooflineadd = und "r ist "v "p  
 "prooflineadd = und hat "r "o "v  
 "prooflineadd = dass "o "v "p ist  
 "prooflineadd = dass "r "o "v hat  
 "prooflineadd = sie "am "p mit "v  
 "prooflineadd = "r ist 'art "o "f  
 "prooflineadd = und "r auch fuer "f  
 "prooflineadd = und fuer "o "v: "f  
 "prooflineadd = "hence zu "f und "f  
 "prooflineadd = faellt "o von "v weg  
 "prooflineadd = so dass dieser "o "am  
 "prooflineadd = "hence "f fuer "p "v  
 "prooflineadd = der "p fall geht analog  
 "prooflineadd = "hence fuer "qt "p "v  
 "prooflineadd = daher bleibt auch "r "p  
 "prooflineadd = ist "hence 'art "p "o  
 "prooflineadd = und "follows "r "o "f

"prooflineadd = und aus "r "follows "f  
 "prooflineadd = und 'because "v "p war  
 "prooflineadd = 'because "v "am in "v  
 "prooflineadd = 'because "v "p "p ist  
 "prooflineadd = ist "hence genau dann "v  
 "prooflineadd = laesst sich "v so waehlen  
 "prooflineadd = und "r "follows "aa "o  
 "prooflineadd = und fuer "f "follows "f  
 "prooflineadd = und fuer "f "follows "r  
 "prooflineadd = "if wir "f zeigen koennen  
 "prooflineadd = und "r "follows durch "o  
 "prooflineadd = und definieren dazu "o "f  
 "prooflineadd = 'because "qt "o "p sind  
 "prooflineadd = "hence muss man "f waehlen  
 "prooflineadd = so "follows "f "since "r  
 "prooflineadd = wir koennen "hence "f "am  
 "prooflineadd = dass "v tatsaechlich "p ist  
 "prooflineadd = und "r "follows "since "f  
 "prooflineadd = und "since "r "follows "f  
 "prooflineadd = "but natuerlich auch fuer "v  
 "prooflineadd = "f "since definition von "v  
 "prooflineadd = "hence "follows "f "o "ql  
 "prooflineadd = der rest "follows "since "r  
 "prooflineadd = "if aus dieser beziehung folgt  
 "prooflineadd = "hence "follows durch "aa "f  
 "prooflineadd = "hence "follows "since "r "p  
 "prooflineadd = dass "n in "r "o ist  
 "prooflineadd = dass "v in "v "p ist  
 "prooflineadd = wo "f mit "p "v "am  
 "prooflineadd = dass "o "f "o "f hat  
 "prooflineadd = dass es "o mit "f gibt  
 "prooflineadd = und "qt "v ist dann "f  
 "prooflineadd = und man kann "p "f "aa  
 "prooflineadd = und nehmen an "v sei "p  
 "prooflineadd = und wendet auf "v "n an  
 "prooflineadd = was sich zu "f "aa "am  
 "prooflineadd = so waere "f fuer "qt "v  
 "prooflineadd = "hence "aa wir in "o "f  
 "prooflineadd = sodass sich "o zu "n "am  
 "prooflineadd = "hence "if "o "v "p ist  
 "prooflineadd = "hence "v und "v sind "p  
 "prooflineadd = "hence hat "o "f "qt "o  
 "prooflineadd = es reicht also "r zu zeigen  
 "prooflineadd = koennen wir "r zum "o "aa  
 "prooflineadd = 'because "f "p ist zu "f  
 "prooflineadd = dass "r nur dann gueltig ist  
 "prooflineadd = so "follows fuer "r "f "r  
 "prooflineadd = so hat man 'art "o gefunden  
 "prooflineadd = so koennen wir "v so waehlen

"prooflineadd = 'because "f mit "f "p ist  
 "prooflineadd = "hence kann "v "qt "o sein  
 "prooflineadd = die man nach "v auflösen kann  
 "prooflineadd = so wäre "f im Widerspruch "r  
 "prooflineadd = und "since "r ist "hence "f  
 "prooflineadd = wie man sofort aus "r entnimmt  
 "prooflineadd = 'because sie schon in "v "am  
 "prooflineadd = müssen "qt "o zu "v gehören  
 "prooflineadd = so können wir wieder "r bilden  
 "prooflineadd = die "p im 'nr "o nachzuprüfen  
 "prooflineadd = wie man ebenso wie "r nachprüft  
 "prooflineadd = so "follows zunehmend "p "o "v  
 "prooflineadd = was "since "f nicht möglich ist  
 "prooflineadd = "hence "follows "f fuer "qt "v  
 "prooflineadd = stimmen "v und "v überall überein  
 "prooflineadd = und dass wird gerade "r ausgedrückt  
 "prooflineadd = was gerade durch "o gewährleistet ist  
 "prooflineadd = so gibt es "o "ql mit "f  
 "prooflineadd = und es gilt fuer ein "v "f  
 "prooflineadd = und "aa "v so "p, dass "f  
 "prooflineadd = und seine "o ist die von "v  
 "prooflineadd = dass "v und "v "p "o haben  
 "prooflineadd = dass die "p "v "o "ql "am  
 "prooflineadd = so fallen "qt "o mit "f weg  
 "prooflineadd = und hat daher sein "o bei "v  
 "prooflineadd = "hence gibt es ein "v mit "f  
 "prooflineadd = "hence ist "v in "v "p: "f  
 "prooflineadd = "if wir "v "p durch "f "am  
 "prooflineadd = "hence ist "v in "v sogar "p  
 "prooflineadd = indem man 'art "p "o "p "aa  
 "prooflineadd = "if man "v als "p "o auffasst  
 "prooflineadd = es gibt "hence 'art "v mit "f  
 "prooflineadd = dass "o mit "p "o zuerst kommen  
 "prooflineadd = so könnten wir "p "o "v finden  
 "prooflineadd = 'because nur "o "f zur "o "am  
 "prooflineadd = "hence kann "o "f "qt "o haben  
 "prooflineadd = indem man "r "v durch "v ersetzt  
 "prooflineadd = und wie zuvor schliesst man auf "f  
 "prooflineadd = wobei "v durch "v zu ersetzen ist  
 "prooflineadd = 'because "qt "o den "o "v "am  
 "prooflineadd = "hence muss "since "r "r "p sein  
 "prooflineadd = so "follows aus "f "aa zunächst "f  
 "prooflineadd = "hence die behauptung gilt auch fuer "v  
 "prooflineadd = "since wahl von "v "follows "hence "f  
 "prooflineadd = 'because die 'nr "o "f ausgeschlossen wurde  
 "prooflineadd = und "aa sie zu "o "v von "v  
 "prooflineadd = und "v "o mit "v "p "o hat  
 "prooflineadd = dass die "p "o "v in "v "am  
 "prooflineadd = "hence ist "v ein "o und "v "o

"prooflineadd = "if man "v durch "v mit "f "aa  
 "prooflineadd = dass 'art "p "o "v mit "f "am  
 "prooflineadd = und nicht von der wahl "o "v "am  
 "prooflineadd = und erhaelt im "o einen "p "o "v  
 "prooflineadd = "hence "am "v "o "ql mit "o "f  
 "prooflineadd = dass ein anderes "o "v mit "v "am  
 "prooflineadd = und sonst laesst sich "o als "o "aa  
 "prooflineadd = "hence gehoert fuer "v "o "v zu "v  
 "prooflineadd = dass sie nach dem "p "o geordnet sind  
 "prooflineadd = und die "r laesst sich weiter "aa "f  
 "prooflineadd = und fuer "f ueberall ausser im "p "o  
 "prooflineadd = 'because 'art "p "o von "v "p ist  
 "prooflineadd = und erhaelt nach "aa "r die andere "o  
 "prooflineadd = "since "f ist das ein widerspruch zu "r  
 "prooflineadd = "hence "follows fuer 'art "p "o "v "f  
 "prooflineadd = "hence koennen wir "v so waehlen, dass "f  
 "prooflineadd = 'because fuer "f entweder "f oder "f ist  
 "prooflineadd = 'because man stets "v mit "f waehlen kann  
 "prooflineadd = nehmen wir an dass "v in "v "p ist  
 "prooflineadd = was fuer "o "v nur fuer "f "p ist  
 "prooflineadd = so kann man "qt "o "f nach "v "aa  
 "prooflineadd = "if "qt "o von "v "qt "o "v "am  
 "prooflineadd = "hence "aa wir "v "o nach wie "r an  
 "prooflineadd = "hence nimmt "o "v auf "v ihr "o an  
 "prooflineadd = 'because "v ein "o mit den "o "v ist  
 "prooflineadd = 'because in "o "v nur "o mit "f "am  
 "prooflineadd = "hence kann "r "qt "p "o mit "f "am  
 "prooflineadd = und die "p "o "r "follows fuer "f "f  
 "prooflineadd = indem wir das "p "o "v als "v schreiben  
 "prooflineadd = 'because "p "o mit "p "o stets "p sind  
 "prooflineadd = "hence dass "qt "o ausreichen, um "v zu "am  
 "prooflineadd = 'because fuer "o "r nur fuer "f verlangt wird  
 "prooflineadd = fuer das die "p "o von "v "o "v "am  
 "prooflineadd = "hence gibt es "n "o "v und "v mit "f  
 "prooflineadd = dass "o "qt "o mit "o wieder zu "o "am  
 "prooflineadd = so "aa man sie durch weitere "o "v zu "o  
 "prooflineadd = indem man eine "p "o 'art "p "p "o "aa  
 "prooflineadd = 'because die "p "p "o von "v die "p ist  
 "prooflineadd = ist fuer feste "v durch "f 'art "p "o "p  
 "prooflineadd = 'because die "v und "v "o zum "o "v sind  
 "prooflineadd = kann man "o auch von "v bis "v laufen lassen  
 "prooflineadd = und wenden "r noch einmal mit "v statt "v an  
 "prooflineadd = "hence kann man "v durch "aa "qt der "r "am  
 "prooflineadd = 'because "o von "f genau dann "qt "am, wenn "f  
 "prooflineadd = 'because "v "o ist, kann "v so "p gewaehlt werden  
 "prooflineadd = "hence ist in "o im "o mit "f "o "v "p  
 "prooflineadd = und fuer "v einen "p zu "v "p "o "o "v  
 "prooflineadd = und "since "f fuer "f liegt der ganze "o in "v  
 "prooflineadd = dass "o bei "o, "f fuer "p "v von "o "v ist

```

"prooflineadd = fuer die aus den "v "p "o gilt fuer "f "r "f
"prooflineadd = "hence hat die durch "f "p "o "v ueber "v "o "f
"prooflineadd = "hence finden wir mit 'art "p "o "f von "v "r "f
"prooflineadd = "if man die wegen "f staerkere "p "o "f fuer "f benutzt
"prooflineadd = und 'because 'art "p "o aus 'art "p "follows "r, "follows
    "r
"prooflineadd = die aus "v durch "o nach der "p "o nach dem "o "am
"prooflineadd = es sei "f und "r ist richtig fuer "o mit nur "v zeilen
"prooflineadd = "hence ist umgekehrt "qt "o "v unter "v auch "o "v unter "v
"prooflineadd = indem man die "p "o "v von "v durch die "p "o "v "aa
"prooflineadd = dass die beiden "o links und rechts von "r ein "p "v auf
    "p "o "am
"prooflineadd = "hence findet man "since der "o 'art "o genau "p "o wie in
    der "p "o
"prooflineadd = 'because im "p "o "o fuer "f der "p "o und fuer "f der "p
    "o "am
"prooflineadd = die aus "v durch "aa 'art "p "o und "aa der "p "o durch
    'art "p "o "am

```

## Part IV

# Bibliography

## References

- [1] Neumaier, A.: *Analysis und lineare Algebra*, Unpublished lecture notes, 2008.  
<http://www.mat.univie.ac.at/~neum/FMathL/ALA.pdf>
- [2] <http://www.mat.univie.ac.at/~neum/FMathL.html#ALA>
- [3] Cohen, Daniel I.: *Introduction to computer theory*, John Wiley & Sons, Inc., New York, 1986.
- [4] *FMathL - Formal mathematical language*,  
<http://www.mat.univie.ac.at/~neum/FMathL.html>
- [5] *GF - Grammatical Framework*,  
<http://www.cs.chalmers.se/~aarne/GF/>
- [6] *NAPROCHE - Natural Language Proof Checking*  
<http://www.math.uni-bonn.de/people/naproche/index.shtml>
- [7] *Mathlang: Experience-driven development of a new mathematical language*, Kamaredine, F. and Maarek, M. and Wells, JB, Electronic Notes in Theoretical Computer Science, 93, Elsevier, 2004.
- [8] *The Attempto Project*  
<http://attempto.ifi.uzh.ch/site/>

- [9] K. Angelov: *Type-Theoretical Bulgarian Grammar*. In B. Nordström and A. Ranta (eds), *Advances in Natural Language Processing (GoTAL 2008)*, LNCS/LNAI 5221, Springer, 2008.