

# Inhaltsverzeichnis

1	Einleitung	4
2	Motivation	8
2.1	Problemstellung: Rekonstruktion von Programmfragmenten objektorientierter Sprachen	9
2.1.1	Analyse imperativer Programmiersprachen	10
2.1.2	Analyse objektorientierter Sprachen	19
2.1.3	Wege zu einem neuen Ansatz	31
2.1.4	Lösungsansätze für nicht berücksichtigte Konstrukte von <b>C++</b> und <b>JAVA</b>	60
2.2	Problemstellung: Hominidenevolution	63
3	Problemformulierung	71
3.1	Polynomieller Ansatz des Erbens über nur einen Ahnen (single inheritance)	71
3.1.1	Interaktiver Ansatz (Informatik)	71
3.1.2	Ansatz zur Stammbaumrekonstruktion (Biologie)	99
3.1.3	Mathematisches Modell	102
3.2	NP-schwere <b>BS<sub>C++</sub></b> -Analyse (multiple inheritance; Informatik)	105
3.2.1	Übertragung auf die Fragmentanalyse	110
3.2.2	Polynomielle Umrechnung	114
3.2.3	Endlichkeitsbetrachtung	126
4	Mathematische Modellierung konsistenzerhaltender Erweiterungen	127
4.1	Exkursion	138
4.1.1	Behandlung der Phantome und Maxima aus Sicht der Informatik	138
4.1.2	Behandlung der Phantome und Maxima aus Sicht der Biologie	139
4.1.3	Reduktion des Problems um die Menge $M$ der maximalen Knoten	139
4.1.4	Reduktion des Problems um die Menge $P$ der Phantome	140
5	Reduktion des Problems vor der Kronen-Korrektur	146
5.1	Vorläufige Korrektur und Axiomatik	146
5.2	Charakterisierung des Betabaumgraphen	148
5.3	Charakterisierung von Kronen	156
6	Kronen-Minoren	164
6.1	Konstruktion von Trennstellen aus halben Trennstellen	177
6.1.1	Nachbessern des Rautenaxioms	184
6.1.2	Beweisskizze zu VERMUTUNG 1	187
6.1.3	Weitere Betrachtungen	190
6.1.4	Beweis der Umwandelbarkeit in einen binären Baum	191
7	Laufzeitverhalten	193
7.1	Pseudocode des Algorithmus	193
7.2	Polynomielle Entscheidbarkeit	195
8	Zusammenfassung und Ausblick	199
9	Anhang	205
9.1	Beispiele für Korrektur	205
9.1.1	Beispiel 1	205
9.1.2	Beispiel 2	209
9.1.3	Beispiel 3	211
9.1.4	Beispiel 4	213
9.2	Begriffserklärungen	214
9.3	Index	217
9.4	Literaturverzeichnis	219