Habilitationsschrift

genehmigt vom

Fachbereich Informatik der Technischen Hochschule Darmstadt

Algorithmisches Lernen von Funktionen und Sprachen

von Dr. rer. nat. Thomas Zeugmann

D 17 Darmstadt 1993

Referent: Prof. Dr. Rüdiger Reischuk
Korreferenten: Prof. Dr. Hermann Walter
Prof. Dr. Britta Schinzel

Datum des Habilitationsvortrages: 25. November 1993

Inhaltsverzeichnis

1. Einleitung 2

2. Präliminarien 12

2.1. Bezeichnungen 12

2.2. Formalisierung des Lernens von Funktionen 13

2.3. Formalisierung des Lernens von Sprachen 20

3. Zuverlässiges Lernen von Funktionen 28

3.1. Abgeschlossenheitseigenschaften 28

3.2. Die Leistungsfähigkeit deterministischer zuverlässiger Lernalgorithmen . . . 32

3.3. Monte-Carlo Inferenz und zuverlässige Häufigkeitserkennung 44

3.4. Die Leistungsfähigkeit zuverlässiger Häufigkeitserkennung 49

3.5. Diskussion der Resultate und offene Probleme 61

3.6. Zusammenfassung 63

4. Algorithmisches Lernen von Sprachen 66

4.1. Lernen bei Textbelehrung 66

4.1.1. Lernen im Limes und finite Inferenz 66

4.1.2. Lernen unter einer begrenzten Anzahl von Hypothesenwechseln . . . 70

4.1.3. Monotones und dual monotones Lernen 81

4.2. Lernen bei Informantbelehrung 100

4.3. Charakterisierungen 108

4.3.1. Charakterisierung konservativer Inferenz 110

4.3.2. Charakterisierung streng-monotonen Lernens 116

4.3.3. Charakterisierung finiten Lernens 121

4.4. Exaktes, klassentreues und klassenumfassendes Lernen 123

4.5. Ausblick 130

5. Literaturverzeichnis 135

1. Einleitung

Ausgangspunkt der vorliegenden Arbeit waren zwei, seit mehreren Jahren ungelöste Probleme im Gebiet des algorithmischen Lernens von Funktionen und Sprachen. Wir wollen daher zunächst eine Einordnung dieser Probleme vornehmen und dabei gleichzeitig den Gegenstand der algorithmischen Lerntheorie erläutern.

Es wäre schön, an dieser Stelle eine Definition von Lernen anzugeben und darauf aufbauend, alles weitere zu entwickeln. Dazu ist es jedoch noch zu früh. Vielmehr muß es, Angluin (1992) folgend, als ein wesentliches Ziel der Forschung angesehen werden, zu einem vertieften Verständnis dessen zu gelangen, was Lernen eigentlich ist. Es ist anderseits unbestritten, daß Inferenzprozesse eine wesentliche Rolle beim Lernen spielen. Zahlreiche Philosophen haben sich in den vergangenen 2000 Jahren intensiv mit induktiver Inferenz beschäftigt und einige ihrer Erkenntnisse und Prinzipien werden gegenwärtig in der algorithmischen Lerntheorie neu durchdacht und beleuchtet (cf., z.B. William of Occam, ca. 1320, Freivalds, 1990, Board und Pitt, 1992 sowie Popper, 1968 und Case und Smith, 1983).

Per definitionem ist induktive Inferenz der Prozeß des Generierens von Hypothesen zur Beschreibung eines unbekannten Objektes aus endlich vielen Daten über das zu erkennende bzw. zu erlernende Objekt. So führte Gallilei beispielsweise am schiefen Turm in Pisa Messungen zum freien Fall durch. Aus den dabei gemessenen Daten (Fallzeit, Fallhöhe etc.) inferierte er dann ein *Gesetz* in Form einer mathematischen Formel, mit dem sich der freie Fall im Vakuum für die meisten Anwendungen hinreichend genau beschreiben läßt. Erkenntnistheoretisch kann die verwendete Methode wie folgt interpretiert werden. Die Fallhöhe x wird als Datum variiert und die resultierende Fallzeit $f(x)$ gemessen. Dabei erhält der Experimentator eine endliche Folge von Paaren $(x_0, f(x_0)),\ (x_1, f(x_1)), ..., (x_n, f(x_n))$. Aus diesen Beispielen wird nun der Zusammenhang zwischen x und $f(x)$ *inferiert*, indem für f ein mathematischer Ausdruck (eine Formel, ein Algorithmus) als Hypothese angenommen wird. Vermöge weiterer Experimente kann die jeweils aktuelle Hypothese entweder bestätigt oder aber falsifiziert werden. Das heißt, nachdem nur ein endlicher Teil der potentiell unendlichen Folge von experimentellen Daten von Gallilei ausgewertet wurde, konvergierte er gegen eine Formel, also eine endliche Beschreibung, die mit allen weiteren Experimenten (unter den gleichen Randbedingungen) verträglich war, die also, wie Popper (1968) sagt, nicht mehr falsifiziert werden konnte. Mit dieser von Galillei gefunden Hypothese wurde dann bis zur Ausarbeitung der speziellen und allgemeinen Relativitätstheorie so gearbeitet, als ob es eine richtige wäre,

obwohl sie tatsächlich nur eine Approximation darstellt. Impliziert wurde von Galilei von Anfang an angenommen, daß es tatsächlich ein Fallgesetz gibt und dieses Gesetz mit den ihm zur Verfügung stehenden mathematischen Instrumentarien ausdrückbar ist. Diese Annahme war, wie wir heute wissen, falsch. Hätten also die damaligen Meßgeräte die erforderliche Genauigkeit besessen, so wäre jede von Galillei hypothetisch angenommene Formel falsifiziert worden. Damit besitzt die von ihm verwendete Inferenzmethode eine Eigenschaft, die wir Zuverlässigkeit nennen wollen und auf die wir später noch ausführlich zu sprechen kommen werden.

Schon bald nach dem Aufkommen der ersten Computer gab es Wissenschaftler, die sich fragten, ob Inferenzprozesse der oben beschriebenen Form nicht auch von Maschinen realisiert werden können. Grundlegende mathematische Formalisierungen finden sich in Putnams (1975) „Voice of America Lecture“ von 1963 sowie in den Arbeiten von Solomonoff (1964) und von Gold (1965, 1967). Im Ansatz von Putnam (1975) und von Gold (1965) sind die zu inferierenden Objekte rekursive Funktionen. Die zur Verfügung stehenden Beispiele bzw. Informationen sind sukzessiv wachsende Anfangsstücke der entsprechenden Funktionsgraphen. Im einfachsten Fall stehen einer *induktiven Inferenzmaschine* M in jedem Schritt $k = 0, 1\,2, ...$ Anfangsstücke $(0, f(0)),\ (1, f(1)), ..., (k, f(k))$ zur Verfügung. Die induktive Inferenzmaschine M bildet aus diesen Beispielen in berechenbarer Art und Weise Hypothesen h_k, die wir als Nummern von Programmen in bestimmten berechenbaren Numerierungen von (allen) partiell-rekursiven Funktionen interpretieren. Dabei fordern wir vom Hypothesenraum stets, daß er für jede zu lernende Funktion mindestens ein Programm enthält. Gilt $h_k \neq h_{k+1}$, so sagen wir, daß M einen Hypothesenwechsel ausgeführt hat. Die produzierte Hypothesenfolge soll nun in Abhängigkeit vom Lernziel gegen eine korrekte Beschreibung der zu erkennenden Funktion konvergieren. Die Korrektheitskriterien können dabei von völlig korrekt bis hin zu endlich vielen Fehlern variieren. Wir werden auch verschiedene Konvergenzbegriffe betrachten. Im einfachsten Fall hat sich die Hypothesenfolge auf ein Programm zu stabilisieren, welches die zu lernende Funktion berechnet (Lerntyp EX^a, cf., Definition 1).

Aus diesem Ansatz heraus sind in der Literatur zahlreiche Inferenzprobleme definiert und studiert worden. So hat sich die induktive Inferenz zu einer gut ausgebauten mathematischen Theorie entwickelt, deren Resultate in Monographien (cf., Barzdin, 1974, 1975, 1977, Lindner und Thiele, 1986), Übersichtsartikeln (cf., Angluin und Smith, 1983, 1987, Daley, 1986, Klette und Wiehagen, 1980) bzw. einführenden Texten (cf., Osherson, Stob und Weinstein, 1986) sowie zahlreichen technischen Artikeln vorliegen. Die erzielten Ergebnisse führten zu einem wesentlich vertieften Verständnis von Inferenz- und Lern-

prozessen, das auch potentielle Anwendungen in der künstlichen Intelligenz aufzeigte (cf., Osherson, Stob und Weinstein, 1986). Darüber hinaus gelang es, erste Inferenzsysteme zu implementieren (cf., z.B. Nix, 1983, 1984, Shapiro, 1982, Shinohara, 1982).

Neben vielen Vorzügen hat das Goldsche Modell allerdings auch Nachteile. Es erwies sich nämlich als sehr schwierig, für Anwendungen relevante Komplexitätsuntersuchungen in diesem Modell durchzuführen (cf., z.B. Pitt, 1989b). Diese Kritik führte Valiant (1984) zu einem neuen Ansatz, der sich in gewisser Weise wie folgt charakterisieren läßt. Einerseits wurde das Goldsche Modell durch Einführung statistischer Parameter angereichert und andererseits durch Hinzunahme starker Effizienzforderungen verschärft. Die in den letzten Jahren in diesem Modell durchgefühten Untersuchungen bereicherten die algorithmische Lerntheorie außerordentlich (cf., Natarajan, 1991). Die Erfahrungen zeigen allerdings auch, daß die Weiterentwicklung der induktiven Inferenz, des Valiantschen Ansatzes sowie die Ausarbeitung weiterer Ansätze nur in ihrer Gesamtheit und Wechselwirkung tatsächlich zu einem grundlegenderen Verständnis des Phänomens Lernen führen können.

Die in dieser Arbeit vorgestellten Ergebnisse sollen hierzu einen Beitrag leisten. Im ersten Teil beschäftigen wir uns ausführlich mit zuverlässigen Lernalgorithmen. Zur Illustration betrachten wir zunächst ein Beispiel. Die Klasse der zu lernenden Funktionen sei die Menge Pol aller Polynome mit nichtnegativen ganzzahligen Koeffizienten. Nun gilt für Polynome ganz allgemein der bemerkenswerte Satz, daß $n+1$ Paare $(x_0, f(x_0)), ..., (x_n, f(x_n))$ ein Polynom f vom Grade kleiner gleich n vollständig bestimmen. Das legt es nahe, folgenden Lernalgorithmus für Polynome zu entwerfen. Bei Eingabe eines Anfangsstücks $(x_0, f(x_0)), ..., (x_k, f(x_k))$ wird zunächst das Interpolationspolynom q_0 zu $(x_0, f(x_0))$ berechnet. Ist $q_0(x_j) = f(x_j)$ für $j = 1, .., k$, so wird q_0 als Hypothese ausgegeben. Andernfalls wird das Interpolationspolynom q_1 zu $(x_0, f(x_0))$, $(x_1, f(x_1))$ bestimmt und $q_1(x_j) = f(x_j)$ für $j = 2, ..., k$ getestet. Verläuft der Test positiv, so wird q_1 als Hypothese ausgegeben. Sonst iterieren wir das Verfahren, d.h. wir berechnen q_2 aus $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$ usw. Ist f nun tatsächlich ein Polynom vom Grade n, so konvergiert der beschriebene Algorithmus gerade gegen q_n. Was passiert nun, wenn wir obigen Algorithmus für eine totale Funktion g nutzen, die gar kein Polynom ist. Dann existiert für jedes Polynom $f \in Pol$ mindestens eine Zahl x derart, daß $g(x) \neq f(x)$ ist. Somit verwirft der beschriebene Lernalgorithmus jede seiner Hypothesen irgendwann, d.h. er führt unendlich viele Hypothesenwechsel durch. In anderen Worten, der Algorithmus konvergiert dann und nur dann, wenn er korrekt lernt. Derartige Lernalgorithmen wollen wir *zuverlässig* nennen (cf., Definiton 3). Nun haben wir bei der Ableitung der

Divergenz der Hypothesenfolge jedoch wesentlich die Totalität von g benutzt. Wäre g eine partielle Funktion, die genau an n Stellen definiert und dort gleich einem Polynom ist, so konvergiert der obige Algorithmus offenbar gegen eine falsche Hypothese. Unser Lernverfahren ist also zuverlässig auf der Menge der totalen Funktionen, jedoch *nicht* zuverlässig auf der Menge der partiellen Funktionen. Deshalb werden wir in der Folge stets von Zuverlässigkeit bzgl. einer Menge $\mathcal{M}$ von Funktionen reden.

Zuverlässige Lernalgorithmen wurden erstmals von Blum und Blum (1975) sowie von Minicozzi (1976) untersucht. Dabei wurde entdeckt, daß sie einige sehr nützliche Eigenschaften besitzen. Kann eine Klasse U_1 durch M_1 und eine Klasse U_2 durch eine Maschine M_2 zuverlässig auf einer Menge $\mathcal{M}$ gelernt werden, so existiert stets eine induktive Inferenzmaschine M, die $U_1 \cup U_2$ zuverlässig auf $\mathcal{M}$ lernt. In gewisser Weise modellieren zuverlässige Lernalgorithmen auch das Phänomen des „Lernens wie gelernt wird“. Intuitiv stellt man sich dabei vor, daß im Lernprozeß gewonnene Erfahrungen beim Lernen komplizierterer Objekte hilfreich eingesetzt werden können (cf., Gasarch und Smith, 1986). Wir könnten die oben betrachtete Klasse der Polynome beispielsweise zur Menge aller derjenigen Funktionen erweitern, die an fast allen Stellen gleich einem Polynom sind. Ist diese Klasse immer noch zuverlässig auf der Menge aller totalen Funktionen erlernbar? Gemäß einem in Minicozzi (1976) gezeigten Satz ist diese Frage bejahend zu beantworten. Ganz allgemein gilt nämlich, daß sich jede auf einer Menge $\mathcal{M}$ zuverlässige induktive Inferenzmaschine M zu einer Maschine $\hat{M}$ modifizieren läßt, die alle endlichen Varianten der von M gelernten Funktionen zuverlässig auf $\mathcal{M}$ inferiert.

Es blieb jedoch lange offen, ob sich der Zuverlässigkeitsbegriff auch auf abgeschwächte Varianten des Goldschen „Lernen im Limes“ verallgemeinern läßt. Die ersten Verallgemeinerungen wurden von Barzdin (1974b) und Podnieks (1974) untersucht. Barzdin (1974b) schwächte die Forderung nach syntaktischer Konvergenz (d.h. gegen *ein* korrektes Programm) durch semantische Konvergenz ab. Von der Folge aller produzierten Hypothesen wird dabei nur noch verlangt, daß fast alle Programme die zu erlernende Funktion richtig berechnen (Lerntyp BC^a, cf., Definition 4). Case und Smith (1983) nannten dieses Art des Lernens verhaltenskorrekt und untersuchten sie systematisch. Podnieks (1974) verlangte von der Folge aller ausgegebenen Hypothesen lediglich, daß in ihr richtige Programme für die zu inferierende Funktion mit einer bestimmten Häufigkeit vorkommen (Lerntyp BC_{freq}, cf., Definition 7). Pitt (1989a) studierte die syntaktische Variante der Häufigkeitserkennung, d.h. eine spezielle und korrekte Hypothese muß mit bestimmter Häufigkeit in der Folge aller Hypothesen auftreten (Lerntyp EX_{freq}, cf., Definition 8). Smith (1982) untersuchte endliche Teams von induktiven Inferenzmaschinen, wobei für

jede Funktion aus der Klasse der zu lernenden Objekte mindestens ein Teammitglied erfolgreich sein muß (Lerntyp EX_{team}, BC_{team}, cf., Definition 10). Wiehagen, Freivalds und Kinber (1984) betrachteten probabilistische induktive Inferenzmaschienen mit syntaktischer Konvergenz. In diesem Ansatz darf die Inferenzmaschine in jedem Berechnungsschritt eine Münze werfen und ihre Berechnung in Abhängigkeit vom Ergebnis des Münzwurfes fortsetzen. Eine probabilistische Inferenzmaschine lernt eine Funktion f mit Wahrscheinlichkeit p, wenn unter allen möglichen Berechnungsverläufen bei sukzessiver Eingabe von $(0, f(0)),\ (1, f(1)), ...$ mit Wahrscheinlichkeit größer oder gleich p Hypothesenfolgen auftreten, die im Limes gegen ein korrektes Programm für f konvergieren. Pitt (1989a) definierte und untersuchte die semantische Variante des probabilistischen Lernens, d.h. in seinem Modell wird eine Funktion f mit Wahrscheinlichkeit p gelernt, wenn unter allen möglichen Berechnungsverläufen mit Wahrscheinlichkeit größer oder gleich p Hypothesenfolgen derart auftreten, daß fast alle in ihnen vorkommenden Programme f berechnen.

Alle diese Verallgemeinerungen haben nun die Gemeinsamkeit, die Lernfähigkeit gegenüber dem Goldschen Ansatz zu vergrößern. Nichtsdestoweniger ist keiner der allgemeineren Lerntypen bzgl. Vereinigung oder endlicher Invarianz abgeschlossen. Deshalb bemühten sich verschiedene Autoren, den Begriff des zuverlässigen Lernens wenigstens in eine der erwähnten Richtungen so zu verallgemeinern, daß zumindest die endliche Vereinigungsstabilität erhalten bleibt. Die dabei gemachten Erfahrungen waren entmutigend. Entweder wurde zu viel gefordert, so daß sich die Lernfähigkeit nicht vergrößerte (cf., Pitt, 1989a) oder die Definitionen waren technisch nur noch schwer zu handhaben. Daley (1986) lenkte deshalb die Aufmerksamkeit erneut auf dieses Problem. Eine vollständige Lösung konnte vom Autor gemeinsam mit E.B. Kinber gefunden werden (cf., Kinber und Zeugmann, 1991). Neben der Verallgemeinerung in *alle* der oben erwähnten Richtungen gelang es auch, die von Pitt (1989a) erzielten Unifizierungen in den Bereich des zuverlässigen Lernens auszudehnen und die Lernfähigkeit der entstehenden Lerntypen transparent zu machen. Die Verallgemeinerung gelang durch folgende Beobachtung. Wenn ein auf einer Menge $\mathcal{M}$ zuverlässiger Lernalgorithmus als Eingabe den Graphen einer Funktion erhält, die er nicht erlernen kann, so muß er unendlich oft die Hypothese wechseln. Somit kann jeder Hypothesenwechsel von einem Nutzer der Inferenzmaschine als Fehlermeldung interpretiert werden. Ist der Inferenzprozeß nicht erfolgreich, so gibt es unendlich viele Fehlermeldungen. Kann die Inferenzmaschine die betrachtete Funktion dagegen erlernen, so produziert sie nur eine endliche Anzahl von Fehlermeldungen. Somit bot es sich an, Inferenzmaschinen zu betrachten, die geordnete Paare (j_n, b) mit $b \in \{0, 1\}$ als Hypothesen ausgeben. Ist $b = 0$, so interpretieren wir $(j_n, 0)$ als Fehlermeldung. In anderen Worten

bedeutet dies, daß eine Ausgabe der Form $(j_n, 1)$ ein Indikator für das Vertrauen der Inferenzmaschine in ihre aktuelle Hypothese darstellt. Um Mißverständnisse zu vermeiden, sei an dieser Stelle noch darauf hingewiesen, daß aus $(j_n, b) = (j_n, 1)$ natürlich nicht die Korrektheit des Programms j_n für f gefolgert werden kann. Erlernt eine Inferenzmaschine eine Funktion aus der Menge $\mathcal{M}$ nicht, so muß sie unendlich viele Fehlermeldungen produzieren. Andernfalls sind fast alle Hypothesen mit einer 1 in der zweiten Komponente versehen. Es macht nun keine Mühe mehr, zuverlässiges verhaltenskorrektes Lernen zu definieren (Lerntyp $\mathcal{M}-RBC^a$, cf., Definition 6) und den Begriff der zuverlässigen Häufigkeitserkennung zu formalisieren (Lerntypen $\mathcal{M}-REX_{freq}$, $\mathcal{M}-RBC_{freq}$, cf., Definition 9). Beim Übertragen dieses Ansatzes auf probabilistische Inferenzmaschinen erhalten wir folgendes. Sei f eine Funktion aus der Menge $\mathcal{M}$, deren Graph sukzessive einer probabilistischen Inferenzmaschine zugeführt wird. Entweder enthalten dann alle Hypothesenfolgen unendlich viele Fehlermeldungen oder alle Hypothesenfolgen enthalten nur endlich viele, unabhängig von der Folge der Münzwürfe. Im zweiten Fall fordern wir nun, daß mit Wahrscheinlichkeit größer gleich p unter allen Hypothesenfolgen syntaktisch oder semantisch konvergente vorkommen. Damit liegt die gesamte Unsicherheit im Bereich des Lernens, so daß wir die entstehenden Lerntypen als *Monte Carlo-Inferenz* interpretieren können (Lerntyp $\mathcal{M}-REX_{prob}$, $\mathcal{M}-RBC_{prob}$, cf., Definition 11). Für eine ausführlichere Diskussion der erzielten Resultate verweisen wir auf Abschnitt 3.5.

Für den 2. Teil der Arbeit hätte es sich natürlich angeboten, zuverlässiges Lernen von Sprachen zu studieren. Wir haben dennoch darauf verzichtet, da es im Gebiet des algorithmischen Lernens von Sprachen andere, seit längerer Zeit offene Probleme gab, deren Lösung unseres Erachtens aus der Sicht potentieller Anwendungen Vorrang gebührte. Ausgangspunkt war ein von Angluin (1980a) gestelltes Problem als auch eine in der KI-Gemeinschaft kontrovers diskutierte Frage.

Beim induktiven Lernen aus Beispielen muß zwangsläufig irgendwann einmal generalisiert werden, da man sonst permanent Gefahr läuft, nicht zu konvergieren. Nicht klar hingegen ist, ob man so stark wie möglich oder so vorsichtig wie gerade nötig generalisieren sollte. Im ersten Fall böte sich dann an, initial die bzw. eine allgemeinste Hypothese zu produzieren und den Lernerfolg anschließend durch Spezialisierung zu realisieren. Die zweite Strategie liefe auf sukzessives Generalisieren hinaus. Schließlich ist es noch denkbar, beide Herangehensweisen zu kombinieren, also durch ein geeignetes Zusammenspiel von Generalisieren und Spezialisieren zum Lernerfolg zu gelangen. Kontrovers diskutiert wurde nun die Frage, welcher dieser Strategien, sofern überhaupt, der Vorzug gebührt (cf., Michalski, Carbonell und Mitchel, 1984, 1986 sowie Kodratoff und Michalski, 1990).

Bei unseren Untersuchungen stellte es sich heraus, daß die Beantwortung dieser Frage im engen Zusammenhang mit dem Problem steht, inwieweit nichtmonotones Denken im Lernprozeß zu modellieren ist. Wir wollen diese Fragen jetzt genauer betrachten.

Zunächst beschreiben wir das allgemeine Modell und arbeiten dabei gleichzeitig Unterschiede und Gemeinsamkeiten zum Lernen rekursiver Funktionen heraus. Die zu inferierenden Objekte sind jetzt Sprachen, die zumindest rekursiv aufzählbar sein sollen. Als Informationen können nur *positive* Beispiele verfügbar sein, d.h. sukzessiv wachsende Anfangsstücke von Folgen t, deren Elemente Wörter der zu lernenden Sprache L sind. Insbesondere wird verlangt, daß jedes Wort aus L mindestens einmal in t vorkommt. Wir nennen t *Text* für L. Die zweite Variante des Informationsangebots besteht darin, daß sukzessive *alle* Wörter über dem zugrunde liegenden Alphabet angeboten werden, wobei jedes Wort aus der zu lernenden Sprache L mit einem $+$ und alle anderen mit einem $-$ markiert sind. Derartige Informationsfolgen nennen wir *Informant*. Synonym sagen wir auch, L wird aus *positiven und negativen* Beispielen gelernt. Wir weisen noch darauf hin, daß wir hinsichtlich der Ordnung, in der die (markierten) Wörter angeboten werden, keine Einschränkungen vornehmen.

Eine induktive Inferenzmaschine berechnet aus den angebotenen Anfangsstücken Hypothesen h, die wir in Abhängigkeit vom gewählten Hypothesenraum interpretieren. Als Hypothesenräume können ganz allgemein berechenbare Numerierungen von Akzeptoren, Grammatiken oder auch charakteristischen Funktionen zugelassen werden. Die Folge aller auf einem Text bzw. Informanten produzierten Hypothesen soll dann wieder in Abhängigkeit vom Lernziel gegen eine korrekte Beschreibung der zu lernenden Sprache konvergieren.

Unterschiede zum Lernen rekursiver Funktionen gibt es hinsichtlich mehrerer Bestimmungsstücke des Modells. Während Funktionsgraphen stets als Informant interpretierbar sind, da aus der Kenntnis von $(x, f(x))$ sofort klar ist, daß $f(x) \neq y$ für alle $y \in \mathbb{N} \setminus \{f(x)\}$, stellen Texte ein Informationsangebot dar, zu dem es im Bereich der induktiven Inferenz von Funktionen kein Analogon gibt. Darüber hinaus haben wir beim Lernen von Sprachen eine größere Variabilität im Lernziel. Für rekursive Funktionen besteht das Lernziel im Prinzip immer darin, Programme zu inferieren, die die zu lernende Funktion bzgl. des gewählten Korrektheitskriteriums berechnen. Das analoge Lernziel für Sprachen ist die Synthese eines Programms für eine charakteristische Funktion der zu inferierenden Sprache. Alternativ ist es jedoch durchaus sinnvoll, eine Sprache als gelernt anzusehen, wenn die Inferenz einer die zu lernende Sprache generierenden Grammatik gelingt. Die beschriebenen Lernziele sind nicht äquivalent (cf., Gold, 1967, Wiehagen, 1978, Case und Lynes, 1982). Es sei noch angemerkt, daß die Forderung aus beliebigen Aufzählun-

gen des Informationsangebots lernen zu müssen, auch im Bereich des Lernens rekursiver Funktionen ausführlich studiert wurde (cf., z.B. Blum und Blum, 1975, Jantke und Beick, 1981). Für die in dieser Arbeit untersuchten zuverlässigen Lerntypen führt diese Bedingung jedoch zu keinerlei Einschränkungen, so daß wir uns stets auf die natürliche Reihenfolge beschränken konnten.

Wir kommen jetzt zur Beschreibung der oben kurz skizzierten Probleme. Dabei schränken wir uns auf das Studium der Lernbarkeit indizierter Familien uniform rekursiver Sprachen ein, da diese aus Sicht potentieller Anwendungen von besonderem Interesse sind. $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ heißt indizierte Familie uniform rekursiver Sprachen, sofern alle L_j nichtleer sind und das Wortproblem uniform für alle L_j entscheidbar ist. Das Lernziel ist im folgenden stets die Synthese von Grammatiken. Ausgehend von Angluins (1980a, 1980b) grundlegenden Arbeiten wurde bislang beim Lernen indizierter Familien $\mathcal{L}$ fast ausschließlich *exaktes* Lernen untersucht, d.h. die indizierten Familien $\mathcal{L}$ sind bzgl. des Hypothesenraumes $\mathcal{L}$ zu lernen. Wir studieren zusätzlich *klassentreues* und *klassenumfassendes* Lernen. Im *klassentreuen* Fall ist als Hypothesenraum jede Numerierung $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ von Grammatiken zugelassen, für die das Wortproblem für alle $L(G_j)$ uniform entscheidbar ist und die der Bedingung $\{L_j \mid j \in \mathbb{N}\} = \{L(G_j) \mid j \in \mathbb{N}\}$ genügt. Wir haben beim Entwurf des Hypothesenraumes also die Freiheit, sowohl die Aufzählung als auch die Beschreibung für die zu lernenden Sprachen geeignet zu wählen. Die Forderung nach Klassentreue ist insofern natürlich, als jede Hypothese, die eine nicht zur Familie gehörende Sprache beschreibt, ohnehin nicht korrekt sein kann. Wie wir später sehen werden, kann es dennoch sinnvoll sein, auch größere Hypothesenräume zuzulassen. Damit wären wir beim *klassenumfassenden* Fall. Hier ist als Hypothesenraum jede Aufzählung von Grammatiken $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ erlaubt, für die das Wortproblem für alle $L(G_j)$ uniform rekursiv ist und die für jede Sprache aus $\mathcal{L}$ mindestens eine Beschreibung enthält. Wir verlangen jedoch nicht mehr, daß jede Grammatik aus $\mathcal{G}$ eine Sprache aus $\mathcal{L}$ generiert.

Wir verfolgen mit unseren Untersuchungen zwei wesentliche Ziele. Einerseits werden wir uns mit der Frage beschäftigen, welchen Einfluß die Wahl des Hypothesenraumes auf die Leistungsfähigkeit von Lernalgorithmen bzw. auf deren Effizienz hat. Diese Problematik studieren wir nun in engem Zusammenhang mit unserem Hauptziel. Dieses besteht darin, den Einfluß unterschiedlicher Monotonieforderungen an den Lernprozeß gründlich zu erforschen. Monotonieforderungen wurden erstmals in Jantke (1991a, 1991b) und Wiehagen (1991) untersucht. Kapur (1992) definierte die dualen Versionen. Die in dieser Arbeit in Kapitel 4 vorgestellten Resultate wurde vom Autor gemeinsam mit S. Lange

und zum Teil mit S. Kapur erzielt (cf., Lange und Zeugmann, 1992a bis 1992d, 1993a bis 1993c, Lange, Zeugmann und Kapur, 1992, Zeugmann, Lange und Kapur, 1992). Inhaltlich geht es darum, ob der Lernprozeß stets so gestaltet werden kann, daß bei der Verarbeitung von mehr und mehr Informationen über die zu inferierende Sprache stets nur „besser“ werdene Generalisierungen bzw. Spezialisierungen als Hypothesen produziert werden. Bei strengster Auslegung dieser Bedingung verlangen wir die Synthese einer Hypothesenfolge, die eine aufsteigende bzw. absteigende Kette von Sprachen ergibt, i.e. $L(G_i) \subseteq L(G_j)$ bzw. $L(G_i) \supseteq L(G_j)$, sofern j nach i ausgegeben wird (cf., Definition 17 und 18, Teil (A)). Die resultierenden Lernmodi nennen wir streng-monoton bzw. dual streng-monoton.

Wiehagen (1991) schlug vor, „besser“ bzgl. der zu lernenden Sprache L zu interpretieren und gab folgende Modifikation an: Anstelle der Forderung $L(G_i) \subseteq L(G_j)$ wird nur noch $L(G_i) \cap L \subseteq L(G_j) \cap L$ verlangt, sofern Hypothese j nach i produziert wird (cf., Definition 17, Teil (B)). Intuitiv gesprochen verlangen wir jetzt von einer monotonen induktiven Inferenzmaschine folgendes. Wird ein Wort w in irgendeinem Lernschritt richtig als zu L zugehörig klassifiziert, so müssen auch die von allen nachfolgenden Hypothesen generierten Sprachen w enthalten. Die duale Begriffsbildung ist natürlich auch für das Spezialisieren sinnvoll. Hier fordern wir von einer induktiven Inferenzmaschine, daß sie alle Wörter, die sie in einem Lernschritt bereits korrekterweise aus L ausgeschlossen hat, auch in den nachfolgenden Hypothesen richtig klassifiziert. Formal ergibt sich damit die Bedingung $co{-}L(G_i) \cap co{-}L \subseteq co{-}L(G_j) \cap co{-}L$ (cf., Definition 18, Teil (B)).

Der dritte Monotonietyp, den wir schwach-monoton bzw. dual schwach-monoton nennen wollen, hat seinen Ursprung im Gebiet nichtmonotoner Logiken. In der klassischen Logik wird stets gefordert wird, daß durch die Hinzunahme zusätzlicher Voraussetzungen zumindest keine Verkleinerung der Menge der daraus ableitbaren bzw. folgerbaren Theoreme resultiert (Monotonieprinzip). Interpretieren wir jetzt die einer induktiven Inferenzmaschine sukzessive zugeführten Informationen über eine zu lernende Sprache als Voraussetzungen, so erkennen wir sofort, daß streng-monotones Lernen gerade dieses klassische Vorgehen modelliert. Nicht monotone Logiken werden vor allem deshalb intensiv untersucht, weil der gesunde Menschenverstand mit den Mitteln der klassischen Logik nur unzureichend zu erfassen ist. Es lag also nahe, auch danach zu fragen, inwieweit nichtmonotones Denken in Lernprozessen zu berücksichtigen ist. Eine sehr natürliche Abschwächung des Monotonieprinzips stellt Kumulativität dar (cf., Brewka, 1991). In kumulativen Logiken muß das Monotonieprinzip erfüllt ein, solange die zu einer Menge V zusätzlich hinzugenommenen Voraussetzungen nicht im Widerspruch zur aus V ab-

leitbaren bzw. folgerbaren Theorie stehen. Kumulativität im Lernprozeß bedeutet somit streng-monotones Lernen, sofern nicht neue Daten vorliegen, die der zuvor produzierten Hypothese widersprechen (cf. Definition 17, Teil (C)). Die resultierende Lernform wollen wir schwach-monoton nennen. Es lag nahe, auch die duale Version schwach-monotonen Lernens zu studieren; hierbei muß dual streng-monoton gelernt werden, solange nicht (markierte) Wörter angeboten werden, die die zuvor ausgegebene Hypothese beweisbar widerlegen (cf. Definition 18, Teil (C)). Schwach-monotones Lernen steht in enger Beziehung zu *konservativem* Lernverhalten, das von Angluin (1980a) eingeführt wurde. Einer *konservativ* arbeitenden induktiven Inferenzmaschine ist es dann und nur dann erlaubt, die Hypothese zu wechseln, wenn die zur Verfügung stehenden (markierten) Wörter der vorherigen Hypothese widersprechen (cf., Definition 15). Wie sich nun heraustellte führt Konservativität zu keiner Einschränkung der Lernfähigkeit von induktiven Inferenzmaschinen, sofern aus positiven und negativen Daten gelernt werden soll. Ist dagegen aus positiven Daten allein zu lernen, so ändert sich die Situation erheblich. Angluin (1980a) konnte dies für den exakten Fall nachweisen. Zur Behandlung des klassentreuen und klassenumfassenden Falls war eine völlig neue Beweistechnik zu entwickeln, die sich auch darüber hinaus als sehr nützlich erwies. Schließlich sei noch darauf hingewiesen, daß es Angluin (1980a) gelungen war, Lernen im Limes aus positiven Daten zu charakterisieren. Diese Charakterisierung erwies sich in vielen Anwendungen als hilfreich. Offen geblieben war hingegen das Problem, ob, und wenn ja wie, konservatives Lernen zu charakterisieren ist. Wir geben in Abschnitt 4.3 eine vollständige Lösung dieses Problems.

Es ist mir ein Bedürfnis, einer Reihe von Freunden und Kollegen an dieser Stelle für die erwiesene Hilfe und Unterstützung zu danken;
Prof. R. Reischuk für die hervorragenden Arbeitsbedingungen, die kreative Atmosphäre sowie für die vielen wervollen Hinweise zur Verbesserung der Darstellung des Themas und das dem Entstehen dieser Arbeit entgegengebrachte Interesse; Prof. E. B. Kinber und Prof. R. Freivalds für die fruchtbare Zusammenarbeit während meines Aufenthalts 1987/88 und danach in Riga; Prof. R. Wiehagen, der mich in die faszinierende Welt der Informatik einführte, für die zahlreichen Diskussionen, Anregungen und Gespräche, die mich oft beflügelt und inspiriert haben; Dr. S. Lange für die zahllosen Tage und Nächte, in denen wir gemeinsam so manches Geheimnis im Bereich des algorithmischen Lernens von Sprachen lüfteten; meinen amerikanischen Kollegen Prof. C. H. Smith, Prof. J. Case, Prof. R. Daley und Prof. M. Velauthapillai für die stets willkommenen und anregenden Diskussionen. Schließlich danke ich Olga, ohne deren Liebe und Verständnis die vorliegende Arbeit nicht hätte entstehen können.

2. Präliminarien

In diesem Kapitel erklären wir zunächst die wichtigsten und im Verlaufe dieser Arbeit oft genutzten Bezeichnungen. Um eine bessere Lesbarkeit zu erreichen, haben wir uns entschieden, selten gebrauchte Bezeichnungen dort einzuführen, wo sie gebraucht werden. Der 2. Teil dieses Kapitels ist der mathematischen Formalisierung der in der Einleitung verbal vorgestellten Lerntypen gewidmet. Dabei erläutern wir einige ihrer wesentlichen Eigenschaften, auf die wir dann in den folgenden Kapiteln zurückgreifen werden.

2.1. Bezeichnungen

Nichtspezifizierte Bezeichnungen werden wie in Rogers (1967) verwendet. Zusätzlich bzw. abweichend von Rogers (1967) nutzen wir folgende Bezeichnungen:

$\mathbb{N} = \{0, 1, 2, ...\}$ bezeichnet die Menge der natürlichen Zahlen. Wir setzen $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Die Klasse aller n–stelligen partiell-rekursiven bzw. allgemein-rekursiven Funktionen über $\mathbb{N}$ sei durch $\mathcal{P}^n$ bzw. $\mathcal{R}^n$ bezeichnet. Ist $n = 1$ so schreiben wir stets kürzer $\mathcal{P}$ bzw. $\mathcal{R}$. Der in Rogers (1967) gemachten Konvention folgend, nennen wir die allgemein-rekursiven Funktionen kurz rekursive Funktionen. Die Klasse aller partiellen bzw. totalen Funktionen über $\mathbb{N}$ bezeichnen wir mit $\mathcal{PF}$ bzw. $\mathcal{TF}$. Sei $f \in \mathcal{PF}$; dann definieren wir $Arg\ f =_{df} \{x \mid f(x) \text{ ist definiert}\}$ und $Val\ f =_{df} \{f(x) \mid f(x) \text{ ist definiert}\}$. Sei jetzt $X \subseteq \mathbb{N}$; mit $\mathcal{TF}_X$ ($\mathcal{R}_X$) bezeichnen wir die Menge aller Funktionen $f \in \mathcal{TF}$ ($f \in \mathcal{R}$), für die $Val\ f \subseteq X$ ist. Insbesondere nehmen wir auf $R_{\{0,1\}}$ oft auch als Menge der rekursiven Prädikate Bezug. Möge nun $n \in \mathbb{N}$ sein. Dann nutzen wir $\mathcal{TF}_n$ und $\mathcal{R}_n$ um die Klasse aller Funktionen $f \in \mathcal{PF}$ bzw. $f \in \mathcal{P}$ zu bezeichnen, für die $card(\mathbb{N} \setminus Arg\ f) \leq n$ ist. Für die Klasse aller Funktionen $f \in \mathcal{PF}$ bzw. $f \in \mathcal{P}$ mit koendlichem Definitionsbereich schreiben wir $\mathcal{TF}_*$ bzw. $\mathcal{R}_*$. Seien $f,\ g \in \mathcal{TF}_*$ und $n \in \mathbb{N}$; dann bedeutet $f(n) \leq g(n)$, daß $f(n)$ und $g(n)$ beide definiert sind und $f(n)$ kleiner oder gleich $g(n)$ ist. Weiterhin nutzen wir die Schreibweise $f =_n g$ und $f =_* g$ genau dann, wenn $card(\{x \mid f(x) \neq g(x)\}) \leq n$ bzw. $card(\{x \mid f(x) \neq g(x)\}) < \infty$. Schließlich schreiben wir $f \subseteq g$, wenn $\{(x, f(x)) \mid x \in Arg\ f\} \subseteq \{(x, g(x)) \mid x \in Arg\ g\}$ gilt.

Mit $\varphi_0,\ \varphi_1,\ \varphi_2, ...$ bezeichnen wir eine beliebig fixierte Gödelnumerierung von $\mathcal{P}$ und mit $\Phi_0,\ \Phi_1,\ \Phi_2, ...$ ein assoziiertes Komplexitätsmaß (cf., Machtey und Young, 1978). Sei $f \in \mathcal{P}$ und $i \in \mathbb{N}$ derart, daß $\varphi_i = f$. Dann nennen wir i ein Programm für f. Ist $\varphi_i(x)$ definiert (Abk. $\varphi_i(x)\downarrow$), so sagen wir des öfteren auch, daß $\varphi_i(x)$ konvergiert. Andernfalls

ist $\varphi_i(x)$ nicht definiert oder divergent (Abk. $\varphi_i(x)\uparrow$). Es ist manchmal sehr zweckmäßig, eine rekursive Funktion mit der Folge ihrer Werte zu identifizieren. So bezeichnet 0^i10^∞ die Funktion f, für die $f(i)=1$ und $f(x)=0$ für alle $x\neq i$ gilt. Unter Verwendung einer beliebig fixierten Kodierung $\langle ...\rangle$ aller endlichen Folgen natürlicher Zahlen auf $\mathbb{N}$ schreiben wir für jedes $n\in\mathbb{N}$ anstelle von $\langle(f(0),...,f(n))\rangle$ kurz f^n, falls $f\in\mathcal{PF}$ und $f(x)$ für jedes $x\leq n$ definiert ist.

Eine Folge $(j_n)_{n\in\mathbb{N}}$ konvergiert im Limes gegen eine Zahl j, falls $j_n=j$ für fast alle n gilt.

Es sei noch vereinbart, daß in der Folge die echte Inklusion stets durch $\subset$ bezeichnet wird, im Unterschied zur einfachen Inklusion $\subseteq$. Die Unvergleichbarkeit von Mengen drücken wir durch $\#$ aus, und die leere Menge bezeichnen wir mit $\emptyset$. Abschließend vereinbaren wir noch, $max\,\emptyset=0$ und $min\,\emptyset=\infty$ zu setzen.

2.2. Formalisierung des Lernens von Funktionen

Wir kommen jetzt zur formalen Definition verschiedener Lerntypen. Sofern nichts anderes gesagt wird, ist eine induktive Inferenzmaschine (Abk. IIM) nichts anderes als eine partiell-rekursive Funktion.

Definition 1. *Sei $a\in\mathbb{N}\cup\{*\}$ und $f\in\mathcal{TF}$. Eine IIM M* ***lernt f im Limes mit maximal a Anomalien****, falls $M(f^n)$ für alle $n\in\mathbb{N}$ definiert ist, und die Folge $(M(f^n))_{n\in\mathbb{N}}$ im Limes gegen eine Zahl j mit $\varphi_j=_a f$ konvergiert.*

Wird f von M mit höchstens a Anomalien im Limes gelernt, so schreiben wir $f\in EX^a(M)$. Die Familie aller im Limes mit maximal a Anomalien erlernbaren Funktionenklassen $U\subseteq\mathcal{R}$ bezeichnen wir mit EX^a, formal $EX^a=\{U\subseteq\mathcal{R}\mid\exists M[U\subseteq EX^a(M)]\}$. Im Falle $a=0$ verzichten wir auf den oberen Index.

Wir weisen noch darauf hin, daß im Fall $a=*$ für jede gelernte Funktion endlich viele Fehler erlaubt sind. Es gibt jedoch im allgemeinen keine für alle gelernten Funktionen geltene uniforme endliche obere Schranke für die Anzahl der Anomalien.

Lernen im Limes wurde 1967 von Gold eingeführt und seitdem intensiv in zahlreichen Modifikationen studiert. Die Bezeichnung EX^a geht auf Case und Smith (1983) zurück und steht für ***ex****plain*. In der Literatur wird der Lerntyp EX^a von einigen Autoren auch mit LIM^a bezeichnet (cf., e.g. Barzdin, 1974, 1975, 1977). Ausgehend von der in der gesamten Wissenschaftsgeschichte gemachten Erfahrung, daß eine wissenschaftliche Theorie nur in den seltensten Fällen völlig fehlerfrei ist, führten Blum und Blum (1975) zuverlässi-

ges Lernen mit endlich vielen Anomalien ein. Dieser Lerntyp spiegelt auch die im Erstellen großer Computerprogramme gemachten Erfahrungen wider, daß ein hinreichend umfangreiches und kompliziertes Programm in den seltensten Fällen bei jeder Eingabe die richtige Ausgabe liefert (cf., Laudan, 1977). Selbst wenn Fehler entdeckt und behoben werden, so hat man noch nicht die Gewißheit, ein fehlerfreies Programm zu besitzen. Damit stellt das Programmieren selbst einen typischen Limesprozeß dar. Case und Smith (1983) verfeinerten nun die Untersuchungen zum Lernen im Limes mit erlaubten Anomalien, indem sie die Leistungsfähigkeit von Lernalgorithmen untersuchten, die im Limes gegen Programme zu konvergieren haben, die eine bestimmte Anzahl $a \in \mathbb{N}$ von Anomalien oder Fehlern nicht überschreiten dürfen. Dabei erzielten sie folgendes Hierarchieresultat:

$$EX \subset EX^1 \subset EX^2 \subset ... \subset \bigcup_{a \in \mathbb{N}} EX^a \subset EX^*$$

Interessante Resultate zur Lernfähigkeit im Limes mit Anomalien finden sich auch in Chen (1982), der diesen Lerntyp mit der Forderung verschärfte, im Limes kürzeste Programme modulo eines rekursiven Faktors für die zu lernende Funktion zu generieren. Royer (1986) untersuchte Verallgemeinerungen des Lerntyps EX^a, wobei er sogar unendlich viele Anomalien zuließ, die jedoch nur auf hinreichend dünnen Teilmengen von $\mathbb{N}$ angenommen werden dürfen. Es blieb allerdings in den genannten Arbeiten offen, welchen Einfluß erlaubte Fehler auf die Leistungsfähigkeit zuverlässiger Infernzmaschinen haben.

Zuverlässige IIMs wurden von Blum und Blum (1975) sowie Minicozzi (1976) eingeführt und können formal wie folgt definiert werden.

Definition 2. *Sei $\mathcal{M} \subseteq \mathcal{TF}$ und $a \in \mathbb{N} \cup \{*\}$. Eine IIM $M \in \mathcal{R}$* ***arbeitet*** $\mathbf{EX^a}$***–zuverlässig auf der Menge $\mathcal{M}$****, falls für alle Funktionen $f \in \mathcal{M}$ gilt: Entweder konvergiert die Folge $(M(f^n))_{n \in \mathbb{N}}$ gegen eine Zahl j mit $\varphi_j =_a f$ oder sie divergiert.*

Eine zuverlässige IIM konvergiert also nur dann bei Eingabe von immer längeren Anfangsstücken des Graphen einer Funktion, wenn sie die Funktion auch erlernt. Mit zuverlässigen IIMs lassen sich jetzt in Analogie zu Definition 1 Lerntypen definieren.

Definition 3. *Sei $\mathcal{M} \subseteq \mathcal{TF}$ und $a \in \mathbb{N} \cup \{*\}$. Eine IIM M* ***erlernt eine Funktion $f \in \mathcal{TF}$ zuverlässig auf der Menge $\mathcal{M}$ mit höchstens a Anomalien****, wenn M auf der Menge $\mathcal{M}$ EX^a–zuverlässig arbeitet und f im Limes mit maximal a Anomalien erlernt.*

Wenn M eine Funktion f zuverlässig auf der Menge $\mathcal{M}$ mit höchstens a Anomalien erlernt, so schreiben wir $f \in \mathcal{M}-REX^a(M)$. Die Familie aller zuverlässig erlernbaren Klassen rekursiver Funktionen wird analog zu Definition 1 erklärt und mit $\mathcal{M}-REX^a$

bezeichnet. Für $a = 0$ lassen wir den oberen Index wieder weg.

Erste Aussagen zur Leistungsfähigkeit zuverlässiger Lernalgorithmen finden sich in Blum und Blum (1975), wo insbesondere $\mathcal{TF}-REX \subset \mathcal{R}-REX \subset EX$ gezeigt und $\mathcal{R}-REX \subset \mathcal{R}-REX^*$ behauptet wurde. Minicozzi (1976) konnte beweisen, daß die Familien $\mathcal{M}-REX^a$ im Falle $a = 0$ und $a = *$ abgeschlossen sind bzgl. rekursiv aufzählbarer Vereinigungen erlernbarer Funktionenklassen als auch gegenüber endlicher Invarianz. Das heißt, wenn eine Klasse $U \subseteq \mathcal{R}$ zuverlässig auf einer Menge $\mathcal{M}$ erlernt werden kann, so ist auch die Klasse aller Funktionen zuverlässig auf $\mathcal{M}$ erlernbar, die fast überall gleich einer Funktion aus U sind. Wir werden weiter unten beweisen, daß alle zuverlässigen Lerntypen diese Abgeschlossenheitseigenschaften besitzen.

Zunächst wollen wir jedoch Definition 1 dahingehend abändern, daß anstelle der syntaktischen Konvergenz der Folge aller generierten Programme die semantische Konvergenz der Hypothesenfolgen gefordert wird.

Definition 4. *Sei $a \in \mathbb{N} \cup \{*\}$ und $f \in \mathcal{TF}$. Eine IIM M* ***erlernt f verhaltenskorrekt mit höchstens a Anomalien****, falls $M(f^n)$ für alle $n \in \mathbb{N}$ definiert ist, und $\varphi_{M(f^n)} =_a f$ für fast alle n gilt. Wir schreiben $f \in BC^a(M)$, falls M die Funktion f verhaltenskorrekt mit höchstens a Anomalien erlernt und setzen $BC^a = \{U \subseteq \mathcal{R} \mid \exists M[U \subseteq BC^a(M)]\}$.*

Verhaltenskorrektes Lernen wurde erstmals von Barzdin (1974b) betrachtet und später von Case und Smith (1983) ausführlich studiert. Die Bezeichnung BC steht für ***b****ehaviorally* ***c***orrect und wurde von Case und Smith (1983) eingeführt. Insbesondere zeigten Case und Smith (1983) folgendes:

$$EX^* \subset BC \subset BC^1 \subset ... \subset \bigcup_{a \in \mathbb{N}} BC^a \subset BC^*$$

Die erste Inklusion wurde schon von Barzdin (1974b) bewiesen. L. Harrington entdeckte nun, daß sogar $\mathcal{R} \in BC^*$ gilt (cf., Case und Smith, 1983). Weitere interessante Resultate zum verhaltenskorrekten Lernen finden sich in Chen (1982), der beispielsweise zeigen konnte, daß die Anzahl der Fehler bei *jeder* IIM, die $\mathcal{R}$ erlernt, von Hypothese zu Hypothese wächst.

Wir kommen jetzt zur Definition von zuverlässiger verhaltenskorrekter Lernbarkeit. Es sei an dieser Stelle noch einmal darauf hingewiesen, daß die nachfolgende Definition aus Definition 4 entstanden ist, indem die für die Zuverlässigkeit wichtigste Eigenschaft in geeigneter Art und Weise modifiziert wurde. Diese Eigenschaft besteht nun gerade darin, die äußere Welt über signifikante Zweifel an der aktuellen Hypothese zu informieren. Bei

der zuverlässigen EX^a–Lernbarkeit *führt die IIM dazu einen Hypothesenwechsel durch.* Bei der zuverlässigen BC^a–Lernbarkeit gibt die IIM nun anstelle von Zahlen geordente Paare (j_n, b) mit $b \in \{0,1\}$ aus. Die erste Komponente wird als Programm interpretiert, während die zweite Komponente die Aufgabe hat, die äußere Welt über eventuelle Fehler zu informieren. So vereinbaren wir, eine Hypothese $(j_n, 0)$ als Fehlermeldung zu verstehen. Ist $b = 1$, so vertraut die IIM ihrer Hypothese.

Definition 5. *Sei $\mathcal{M} \subseteq \mathcal{TF}$ und $a \in \mathbb{N} \cup \{*\}$. Eine IIM $M \in \mathcal{R}$* ***arbeitet* $\mathbf{BC^a}$*–zuverlässig auf $\mathcal{M}$****, wenn für jede Funktion $f \in \mathcal{M}$ gilt: Entweder genügt die von M produzierte Ausgabefolge $((j_n, b_n))_{n\in\mathbb{N}}$ den Bedingungen $(j_n, b_n) = (j_n, 1)$ und $\varphi_{j_n} =_a f$ für fast alle n, oder es gibt unendlich viele n, so daß $(j_n, b_n) = (j_n, 0)$ ist.*

Intuitiv gesprochen erlernt damit eine BC^a–zuverlässige Maschine eine Funktion aus $\mathcal{M}$ verhaltenskorrekt mit höchstens a Anomalien oder sie produziert unendlich viele Fehlermeldungen. Es sei an dieser Stelle noch auf folgenden wichtigen Sachverhalt hingewiesen. Wir *fordern nicht*, daß $(j_n, 1)$ die Gleichheit $\varphi_{j_n} =_a f$ impliziert.

Definition 6. *Sei $\mathcal{M} \subseteq \mathcal{TF}$ und $a \in \mathbb{N} \cup \{*\}$. Eine IIM M* ***erlernt eine Funktion $f \in \mathcal{TF}$ auf der Menge $\mathcal{M}$ $\mathbf{BC^a}$–zuverlässig mit höchstens a Anomalien****, falls M BC^a–zuverlässig auf $\mathcal{M}$ ist und M die Funktion f verhaltenskorrekt mit höchstens a Anomalien erlernt. Wird f von M BC^a–zuverlässig auf $\mathcal{M}$ erlernt, so schreiben wir $f \in \mathcal{M}-RBC^a(M)$. Der Lerntyp $\mathcal{M}-RBC^a$ wird analog wie oben definiert.*

Wir merken noch an, daß man die Fehlermeldungen auch in die aktuellen Hypothesen hinein kodieren könnte, ohne damit den Lerntyp zu verändern. Allerdings wird die Definition dann sehr unhandlich, so daß sich viele Beweise nur mit ungleich höherem technischen Aufwand durchführen lassen.

Mit den nachstehenden Definitionen betrachten wir weitere Varianten des Lernens mit einer gewissen Unsicherheit. Anstelle einer bzw. beliebig vieler korrekter Hypothesen fordern wir jetzt nur noch, daß mit hinreichend großer Häufigkeit korrekte Hypothesen vorkommen bzw. eine korrekte Hypothese mit hinreichend großer Häufigkeit auftritt. Der folgende Lerntyp wurde von Podnieks (1974) eingeführt.

Definition 7. *Sei $0 < p \leq 1$ und $f \in \mathcal{TF}$. Eine IIM M* ***erkennt f verhaltenskorrekt mit Häufigkeit p****, falls $M(f^n)$ für alle $n \in \mathbb{N}$ definiert ist und*

$$\liminf_{k\to\infty} \frac{card(\{n \mid \varphi_{M(f^n)} = f,\ 0 \leq n \leq k\})}{k} \geq p$$

Wir setzen dann $BC_{freq}(p) = \{U \subseteq \mathcal{R} \mid \exists M[M \text{ erlernt jedes } f \in U \text{ mit Häufigkeit } p]\}$.

Podnieks (1974, 1975) konnte nun $BC_{freq}(\frac{1}{n+1}) \subset BC_{freq}(\frac{1}{n+2})$ für alle $n \in \mathbb{N}$ zeigen.

Darüber hinaus entdeckte er, daß die BC_{freq} Hierarchie diskret ist. Genauer, sei p mit $1/n \geq p > 1/(n+1)$ gegeben. Dann ist $BC_{freq}(p) = BC_{freq}(\frac{1}{n})$. Pitt (1989a) definierte das EX–Analogon zu Podnieks verhaltenskorrekter Häufigkeitserkennung und bewies einen analogen Satz.

Definition 8. *Sei $0 < p \leq 1$ und $f \in \mathcal{TF}$. Eine IIM* ***erlernt f im Limes mit Häufigkeit p****, falls $M(f^n)$ für alle $n \in \mathbb{N}$ definiert ist und es eine ausgezeichnete Hypothese j derart gibt, daß*

$$\liminf_{k\to\infty} \frac{card(\{n \mid M(f^n) = j,\ 0 \leq n \leq k\})}{k} \geq p$$

und $\varphi_j = f$.

Wir bezeichnen die Familie aller mit Häufigkeit p im Limes erlernbaren Funktionenklassen mit $EX_{freq}(p)$.

Wir definieren jetzt zuverlässige Häufigkeitserkennung. Dabei fordern wir von den IIMs erneut, geordnete Paare (j_n, b) mit $b \in \{0, 1\}$ auszugeben. Der Unterschied zwischen zuverlässigem verhaltenskorrektem Lernen mit Häufigkeit p und zuverlässiger syntaktischer Häufigkeitserkennung leitet sich aus den Definitionen 7 und 8 ab. Beim zuverlässigen verhaltenskorrektem Lernen auf einer Menge $\mathcal{M}$ müssen für jede Funktion, die gelernt wird, mit Häufigkeit p richtige Programme in der Folge aller produzierten Hypothese vorkommen. Darüber hinaus darf die Hypothesenfolge insgesamt nur endlich viele Paare der Form $(j_n, 0)$ enthalten. Wird eine Funktion aus $\mathcal{M}$ hingegen nicht mit Häufigkeit p verhaltenskorrekt gelernt, so hat die ausgegebene Hypothesenfolge unendlich viele Fehlermeldungen zu enthalten. Im syntaktischen Fall wird die Forderung „mit Häufigkeit p richtige Programme“ durch „ein richtiges Programm mit Häufigkeit p“ ersetzt. Formal erhalten wir:

Definition 9. *Sei $\mathcal{M} \subseteq \mathcal{TF}$ und $0 < p \leq 1$. Eine IIM $M \in \mathcal{R}$* ***erlernt eine Funktion $f \in \mathcal{R}$ auf $\mathcal{M}$ BC–zuverlässig*** *[**EX–zuverlässig**]* ***mit Häufigkeit p****, falls folgende Bedingungen erfüllt sind:*

(1) *Für jedes $g \in \mathcal{M}$ enthält die Hypothesenfolge $(M(g^n))_{n\in\mathbb{N}}$ entweder nur endlich viele Fehlermeldungen und M lernt g verhaltenskorrekt [im Limes] mit Häufigkeit p oder in $(M(g^n))_{n\in\mathbb{N}}$ kommen unendlich viele Fehlermeldungen vor.*

(2) *M lernt die Funktion f verhaltenskorrekt [im Limes] mit Häufigkeit p.*

Wenn f auf der Menge $\mathcal{M}$ von der IIM M BC–zuverlässig [EX–zuverlässig] mit Häufigkeit p erlernt wird, so schreiben wir $f \in \mathcal{M}-RBC_{freq}(p)(M)$ [$f \in \mathcal{M}-REX_{freq}(p)(M)$].

Die resultierenden Lerntypen werden analog wie oben definiert und mit $\mathcal{M}-RBC_{freq}(p)$ *und* $\mathcal{M}-REX_{freq}(p)$ *bezeichnet.*

Eine weitere Möglichkeit für pluralistisches Lernen wurde von Smith (1982) gefunden. Anstelle einer einzelnen IIM M wird jetzt ein Team $M_0, ..., M_{n-1}$ von n Inferenzmaschinen zum Erlernen einer Funktion genutzt. Jedes Teammitglied erhält sukzessive den Graphen der zu erlernenden Funktion als Eingabe und produziert seine eigene Hypothesenfolge. Ist wenigstens eine IIM im Team erfolgreich, so gilt die Funktion als vom Team gelernt. Dabei unterscheiden wir wieder zwischen Lernen im Limes und verhaltenskorrekter Erkennung.

Definition 10. *Sei* $n \in \mathbb{N}$, $n \geq 1$ *und* $U \subseteq \mathcal{R}$. ***Ein Team*** $(M_0, ..., M_{n-1}) \in \mathcal{R}^n$ ***von IIMs lernt*** U ***verhaltenskorrekt*** [***im Limes***]*, falls für jedes* $f \in U$ *eine IIM* $M \in \{M_0, ..., M_{n-1}\}$ *derart existiert, daß* $f \in BC(M)$ $[f \in EX(M)]$ *ist.*

Wird die Klasse U *vom Team* $(M_0, ..., M_{n-1})$ *verhaltenskorrekt* [*im Limes*] *erlernt, so schreiben wir* $U \subseteq BC(M_0, ..., M_{n-1})$ $[U \subseteq EX(M_0, ..., M_{n-1})]$. *Wir setzten* $BC_{team}(n) = \{U \subseteq \mathcal{R} \mid \exists (M_1, ..., M_n)[U \subseteq BC(M_1, ..., M_n)]\}$ *und definieren* $EX_{team}(n)$ *analog.*

Smith (1982) bewies, daß $BC_{team}(n+1) \subset BC_{team}(n+2)$ sowie $EX_{team}(n+1) \subset EX_{team}(n+2)$ für alle $n \in \mathbb{N}$ gilt. Intuitiv erklärt sich die höhere Leistungsfähigkeit von Teams aus der Vereinigungsinstabilität der Lerntypen EX und BC. Bereits Barzdin (1974b) gab Klassen U_1, U_2 mit U_1, $U_2 \in EX$ und $U_1 \cup U_2 \notin BC$ an. In einem Team von 2 Inferenzmaschinen kann die erste Maschine nun alle Funktionen aus U_1 erlernen und die andere IIM die in U_2 enthaltenen Funktionen. Verallgemeinerungen von Definition 10 finden sich in Pitt und Smith (1989), wo Teaminferenz in der Form studiert wurde, daß $k > 1$ IIMs eines Teams $(M_0, ..., M_{n-1})$ erfolgreich lernen müssen. Wie wir unten beweisen werden (cf., Satz 3.1), sind alle oben definierten zuverlässigen Lerntypen vereinigungsstabil, so daß es keinen Sinn macht, Teams von zuverlässigen Maschinen zu betrachten. Interessanterweise gilt im nicht zuverlässigen Fall $BC_{freq}(\frac{1}{n}) = BC_{team}(n)$ und $EX_{freq}(\frac{1}{n}) = EX_{team}(n)$ für alle $n \geq 1$ (cf., Pitt, 1989a). Das heißt, beide Arten des pluralistischen Lernes sind von gleicher Leistungsfähigkeit und es macht somit keinen Unterschied, auf welche Art und Weise ein begrenzter Nichdeterminismus in den Lernprozeß eingeführt wird. Beim zuverlässigen Lernen ergibt sich dagegen ein völlig anderes Bild, wie wir in Kapitel 3 zeigen werden.

Wir wollen diesen Abschnitt mit der Definition von Monte-Carlo Inferenz fortsetzen. Dazu führen wir zunächst informal den Begriff der probabilistischen IIM ein. Bislang waren induktive Inferenzmaschinen stets partiell-rekursive Funktionen, die für jede Eingabe f^n, $n \in \mathbb{N}$, einer zu lernenden Funktion f definiert zu sein hatten. Alternativ kann man

sich IIMs äquivalenterweise auch als limes-rekursive Funktionale definieren, die durch 3–Band Turing-Maschinen T wie folgt realisiert werden. T besitzt ein Eingabeband, auf dem nur gelesen werden darf. Auf diesem Band seien alle Funktionswerte $f(0)$, $f(1)$, ... der zu lernenden Funktion eingetragen. Weiterhin verfügt T über ein Arbeitsband mit Lese-Schreibkopf und ein Ausgabeband, auf dem nur geschrieben werden darf. Die Maschine T liest jetzt, bei unbegrenzter Arbeitszeit, sukzessive mehr und mehr Funktionswerte auf dem Eingabeband, führt Berechnungen auf dem Arbeitsband durch, und schreibt von Zeit zu Zeit Zahlen auf das Ausgabeband. Die so sukzessive produzierte Ausgabefolge ist dann die Folge der auf f produzierten Hypothesen, wobei wir o.B.d.A. annehmen können, daß diese Folge unendlich ist.

Eine probabilistische IIM P wird nun ebenfalls als limes-rekursives Funktional mit folgenden Modifikationen definiert. Die Maschine P besitzt ein zusätzliches Band mit Lesekopf, auf dem eine zufällige, unendliche Folge von Nullen und Einsen eingetragen ist. Die Berechnung der Hypothesen hängt jetzt sowohl von der auf dem Eingabeband stehenden Funktion f als auch von der auf dem zusätzlichen Band befindlichen $0-1$ Folge S ab. Somit arbeitet P bei jeder $0-1$ Folge S wie eine deterministische IIM, die wir mit P^S bezeichnen. Es sei noch vereinbart, die von P^S auf einer Funktion f produzierte Hypothesenfolge in Analogie zu den oben verwendeten Notationen mit $(P^S(f^n))_{n\in\mathbb{N}}$ zu bezeichnen.

Man kann nun zeigen, daß für $ID \in \{EX^a, BC^a\}$ sowie für die unten definierten Lerntypen $\mathcal{M}-REX_{prob}$ und $\mathcal{M}-RBC_{prob}$ die Menge $\mathcal{F}$ aller derjenigen $0-1$ Folgen, für die die Maschine P auf einer Folge $S \in \mathcal{F}$ eine im Sinne von ID erfolgreiche Hypothesenfolge produziert, stets bzgl. des Borelschen Maßes Pr meßbar ist. Das Maß Pr ist anschaulich folgendermaßen definiert. Sei $n \geq 1$, $\alpha = s_0, ..., s_{n-1}$, $s_i \in \{0,1\}$, $0 \leq i \leq n-1$. Dann wird der Menge aller Folgen $S \in \{0,1\}^\infty$, die α als Präfix haben das Maß $Pr[S] = \frac{1}{2^n}$ zugeordnet. Die Wahrscheinlichkeit mit der eine probabilistische IIM P eine Funktion f im Sinne von ID lernt bezeichnen wir dann mit $Pr[P \text{ lernt } f \text{ im Sinne von } ID]$ und definieren sie als $Pr[\{S \in \{0,1\}^\infty \mid P^S \text{ lernt } f \text{ im Sinne von } ID\}]$, also als $Pr[\mathcal{F}]$. Für eine ausführliche Darstellung der hier skizzierten wahrscheinlichkeitstheoretischen Grundlagen verweisen wir auf Pitt (1989a). Wir kommen jetzt zur angekündigten Definition von Monte-Carlo Lerntypen. Monte-Carlo IIMs sind nichts anderes als probablistische IIMs, die nur einen Typ von Fehlern machen.

Definition 11. *Sei $\mathcal{M} \in \mathcal{TF}$, $0 < p \leq 1$ und $f \in \mathcal{R}$.* ***Eine Monte-Carlo Inferenzmaschine P erkennt die Funktion f verhaltenskorrekt [im Limes] auf $\mathcal{M}$ mit Wahrscheinlichkeit p,*** *falls folgende Bedingungen erfüllt sind:*

(1) *Für jedes* $g \in \mathcal{M}$ *gilt entweder* (α) *oder* (β).

(α) *Für alle Folgen* $S \in \{0,1\}^\infty$ *von Münzwürfen enthält die Hypothesenfolge* $(P^S(g^n))_{n\in\mathbb{N}}$ *nur endlich viele Fehlermeldungen und die Wahrscheinlichkeit dafür, daß* P *die Funktion* g *verhaltenskorrekt* [*im Limes*] *erkennt ist größer gleich* p.

(β) *Für alle Folgen* $S \in \{0,1\}^\infty$ *von Münzwürfen kommen in der Hypothesenfolge* $(P^S(g^n))_{n\in\mathbb{N}}$ *unendlich viele Fehlermeldungen vor.*

(2) *Für die Funktion* f *ist Bedingung* (α) *erfüllt.*

Erkennt die Monte-Carlo IIM P *die Funktion* f *mit Wahrscheinlichkeit* p *verhaltenskorrekt* [*im Limes*] *auf* $\mathcal{M}$, *so schreiben wir* $f \in \mathcal{M}-RBC_{prob}(p)(P)$ [$f \in \mathcal{M}-REX_{prob}(p)(P)$]. *Die resultierenden Lerntypen* $\mathcal{M}-RBC_{prob}(p)$ *bzw.* $\mathcal{M}-REX_{prob}(p)$ *sind dann wie üblich definiert.*

Wir merken noch an, daß unsere Monte-Carlo IIMs dann sicher keinen Fehler machen, wenn sie nicht in der Lage sind, eine Funktion $f \in \mathcal{M}$ mit Wahrscheinlichkeit größer gleich p im verlangten Sinne zu lernen. In diesem Fall produzieren sie nämlich stets unendlich viele Fehlermeldungen. Die Beziehungen zwischen zuverlässiger Häufigkeitserkennung und Monte-Carlo Inferenz studieren wir im Abschnitt 3.3.

Alle bislang eingeführten Lerntypen untersuchen wir ausführlich in Kapitel 3. Doch zunächst formalisieren wir im nächsten Abschnitt das Lernen von Sprachen.

2.3. Formalisierung des Lernens von Sprachen

Rein formal könnten wir das Lernen von Sprachen als Inferenz partiell-rekursiver Funktionen definieren. Aus unserer Sicht ist es jedoch für das intuitive Verständnis günstiger, in der Folge die Terminologie der Theorie formaler Sprachen zu verwenden (cf., Bucher und Maurer, 1984). Mit Σ bezeichnen wir ein beliebig fixiertes endliches Alphabet. Das freie Monoid über Σ bezeichnen wir mit Σ^*. Die Elemente von Σ^* heißen Wörter. Die Länge eines Wortes $w \in \Sigma^*$ bezeichnen wir mit $|w|$. Jede Teilmenge $L \subseteq \Sigma^*$ nennen wir Sprache. Mit $co-L$ bezeichnen wir das Komplement von L, i.e. $co-L = \Sigma^* \setminus L$. Möge jetzt L eine Sprache und $t = w_0, w_1, w_2, ...$ eine unendliche Folge von Wörtern mit $Val\ t =_{df} \{w_k \mid k \in \mathbb{N}\} = L$ sein. Dann nennen wir t ***Text*** oder synonym auch ***positive*** Darstellung von L. Mit $Text(L)$ bezeichnen wir die Menge aller Texte von L. Sei jetzt $i = (w_0,b_0),(w_1,b_1),...$ eine unendliche Folge von Elementen aus $\Sigma^* \times \{+,-\}$ derart, daß $Val\ i =_{df} \{w_k \mid k \in \mathbb{N}\} = \Sigma^*$, $i^+ =_{df} \{w_k \mid (w_k,b_k) = (w_k,+), k \in \mathbb{N}\} = L$

und $i^- =_{df} \{w_k | (w_k, b_k) = (w_k, -), k \in \mathbb{N}\} = co-L$. Dann nennen wir i ***Informant*** für L. Wird L durch einen Informanten klassifiziert, so nutzen wir synomym auch die Sprechweise, daß L durch ***positive und negative Daten*** dargestellt wird. Mit $Info(L)$ bezeichnen wir die Menge aller Informanten von L. Mögen jetzt t, i ein Text bzw. Informant und $x \in \mathbb{N}$ sein. Dann bezeichnen t_x, i_x die Anfangsstücke von t bzw. i der Länge $x+1$, so ist beispielsweise $i_2 = (s_0, b_0), (s_1, b_1), (s_2, b_2)$. Weiterhin sei noch vereinbart, für den Wertebereich endlicher Anfangsstücke von Texten und Informanten nachstehende Bezeichnungen zu verwenden. Seien t und i ein Text bzw. Informant und $x, y \in \mathbb{N}$ mit $x \leq y$; dann definieren wir $t_x^+ =_{df} \{s_k | k \leq x\}$, $i_x^+ =_{df} \{s_k | (s_k, +) \in i, k \leq x\}$ und $i_x^- =_{df} \{s_k | (s_k, -) \in i, k \leq x\}$. Schließlich nutzen wir noch die Schreibweise $t_x \sqsubseteq t_y$ ($t_x \sqsubset t_y$) bzw. $i_x \sqsubseteq i_y$ ($i_x \sqsubset i_y$), falls t_x bzw. i_x ein (echter) Präfix von t_y bzw. i_y ist.

Für viele Beweise ist es sehr hilfreich, sich auf auf bestimmte, genormte Informanten bzw. Texte beziehen zu können. Unter dem ***lexikographisch geordneten*** Informanten einer Sprache L verstehen wir denjenigen Informanten für L, bei dem die Wörter aus Σ^* zunächst ihrer Länge nach und Wörter gleicher Länge lexikographisch geordnet auftreten. Für nicht leere Sprachen L führen wir den Begriff des ***kanonisch geordneten*** Textes t^L wie folgt ein. Möge $w_0, w_1, ...$ der lexikographisch geordnete Text von Σ^* sein. Teste zunächst sequentiell, ob $w_z \in L$ für $z = 0, 1, 2, ...$, solange bis das erste z mit $w_z \in L$ gefunden wird. Da $L \neq \emptyset$ muß ein derartiges z stets gefunden werden. Setze $t_0^L = w_z$. Wir setzen die Definition induktiv fort.

$$t_{x+1}^L =_{df} \begin{cases} t_x^L w_{z+x+1}, & \text{falls} \quad w_{z+x+1} \in L, \\ t_x^L w, & \text{sonst, wobei } w \text{ das letzte Wort in } t_x^L. \end{cases}$$

Wie bereits in der Einleitung erwähnt, wollen wir uns in dieser Arbeit ausschließlich mit dem algorithmischen Lernen indizierter Familien uniform rekursiver Sprachen beschäftigen. Angluin (1980a) folgend, nennen wir eine unendliche Folge L_0, L_1, L_2, ... ***indizierte Familie*** uniform rekursiver Sprachen, falls alle $L_j \neq \emptyset$ sind und es eine rekursive Funktion f derart gibt, daß für alle $j \in \mathbb{N}$ und alle Wörter $w \in \Sigma^*$ gilt:

$$f(j, w) = \begin{cases} 1, & \text{falls} \quad w \in L_j, \\ 0, & \text{sonst.} \end{cases}$$

Es sei noch vereinbart, indizierte Familien uniform rekursiver Sprachen im weiteren Verlauf kürzer indizierte Familien zu nennen. Als Beispiel für eine indizierte Familie betrachten wir die Menge $\mathcal{L}$ aller kontext-sensitiven Sprachen über Σ in kanonischer Aufzählung

(cf., Bucher and Maurer, 1984). Wir werden weiterhin in der Folge eine indizierte Familie und ihren Wertebereich meist mit dem gleichen Symbol $\mathcal{L}$ bezeichnen, da aus dem Kontext stets klar hervorgeht, was gemeint ist.

An unsere obigen Bemerkungen anknüpfend wollen wir, wenn es um das algorithmische Lernen von Sprachen geht, IIMs stets als limes-rekursive Funktionale auffassen. Auf dem Eingabeband stehen jetzt ein Text t bzw. Informant i der zu lernenden Sprache. Es sei weiterhin vereinbart, die nach dem Lesen eines Anfangstückes t_x bzw. i_x produzierte Hypothese j_x zu nennen. Wir verlangen von einer IIM *nicht* mehr, daß sie nach jedem Lesen eines Eingabeworts bzw. eines markierten Eingabeworts eine Hypothese auf ihr Ausgabeband schreibt. Warum diese technische Bedingung sinnvoll und nötig ist erläutern wir im Anschluß an Definition 18.

An dieser Stelle haben wir noch zu klären, welche Hypothesenräume wir beim Lernen von Sprachen betrachten wollen. Gold (1967) und Wiehagen (1978) entdeckten, daß die Leistungsfähigkeit von Lernalgorithmen im allgemeinen davon abhängt, ob im Limes Grammatiken, d.h. Generatoren, oder Programme für charakteristische Funktionen der zu inferierenden Sprachen gelernt werden sollen. Case und Lynes (1982) studierten dieses Phänomen ausführlich. Dabei stellte sich heraus, daß prinzipiell mehr lernbar ist, wenn Grammatiken zu inferieren sind. Wie auch immer, im Bereich des Lernens indizierter Familien sind beide Konzepte gleichwertig, solange für alle Sprachen im Hypothesenraum die uniforme Entscheidbarkeit des Wortproblems gesichert ist. Wir haben uns dafür entschieden, als Hypothesen Grammatiken ausgeben zu lassen, da dieses Konzept die Formalisierung von Monotonieforderungen in einer für Sprachen adäquaten Weise erlaubt, es weiterhin die intuitive Seite des Lernens von Sprachen besser widerspiegelt und schließlich einen nahtlosen Übergang zu weiterführenden Untersuchungen ermöglicht (c.f., Abschnitt 4.5.). Möge jetzt $\mathcal{L}$ eine indizierte Familie sein. Solange nichts anderes gesagt ist, seien im folgenden ***Hypothesenräume*** für $\mathcal{L}$ stets rekursive Aufzählungen $G_0, G_1, G_2, \ldots$ von Grammatiken über dem Terminalalphabet Σ mit uniform entscheidbarem Wortproblem, die der Bedingung $\mathcal{L} \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$ genügen. Die IIMs geben dann als Hypothesen Zahlen j aus, die wir als G_j interpretieren. Wenn wir eine indizierte Famile $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ selbst als Hypothesenraum nutzen, interpretieren wir die Indizes j stets auch als Grammatiken, ohne daß es dadurch zu Mißverständnissen kommt, da aus dem Kontext stets klar hervorgeht, was gemeint ist. Wir kommen jetzt zur Definition der in der Einleitung informal erklärten Sprachlerntypen. Die folgende Definition geht auf Gold (1967) zurück und formalisiert den Begriff des Lernens im Limes für Sprachen.

Definition 12. *Seien $\mathcal{L}$ eine indizierte Familie von Sprachen und $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum.* ***$\mathcal{L}$ heißt im Limes bei Textbelehrung [Informantbelehrung] bzgl. $\mathcal{G}$ erlernbar*** *(Bez. $\mathcal{L} \in LIM{-}TXT$, $\mathcal{L} \in LIM{-}INF$), falls eine IIM M existiert, so daß für alle $L \in \mathcal{L}$ und alle Texte $t \in Text(L)$ [Informanten $i \in Info(L)$] gilt*

(1) *M produziert auf t [i] unendlich viele Hypothesen,*

(2) *Die Folge $(M(t_x))_{x\in\mathbb{N}}$ [$(M(i_x))_{x\in\mathbb{N}}$] konvergiert im Limes gegen eine Zahl j mit $L = L(G_j)$.*

Wir sagen von einer indizierten Familie $\mathcal{L}$, daß sie ***exakt*** erlernbar ist, wenn $\mathcal{L}$ bzgl. des Hypothesenraumes $\mathcal{L}$ gelernt werden kann. Den resultierenden Lerntyp bezeichnen wir mit $ELIM{-}TXT$ bzw. $ELIM{-}INF$. Ist $\mathcal{L}$ bzgl. eines Hypothesenraumes $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ mit $\{L(G_j) \mid j \in \mathbb{N}\} = \mathcal{L}$ inferierbar, so nennen wir $\mathcal{L}$ ***klassentreu*** erlernbar und verzichten auf einen Präfix. Schließlich heißt $\mathcal{L}$ ***klassenumfassend*** erlernbar, falls $\mathcal{L}$ bzgl. eines Hypothesenraumes $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ mit $\{L(G_j) \mid j \in \mathbb{N}\} \supseteq \mathcal{L}$ inferiert werden kann. Wir bezeichnen den resultierenden Lerntyp mit $KLIM-TXT$ bzw. $KLIM-INF$. Diese Schreibweisen wollen wir analog bei allen im weiteren definierten Lerntypen verwenden.

Anhand der von einer IIM produzierten Folge von Hypothesen kann im allgemeinen nicht entschieden werden, ob eine IIM ihre Lernaufgabe bereits erfolgreich gelöst hat. Wird diese Entscheidbarkeit zusätzlich verlangt, so sprechen wir von finitem Lernen. Finites Lernen wurde erstmals in Gold (1967) formalisiert und ausführlich von Trakhtenbot und Barzdin (1970) studiert.

Definition 13. *Seien $\mathcal{L}$ eine indizierte Familie und $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum.* ***$\mathcal{L}$ heißt finit bei Textbelehrung [Informantbelehrung] bzgl. $\mathcal{G}$ erlernbar*** *(Bez. $\mathcal{L} \in FIN{-}TXT$, $\mathcal{L} \in FIN{-}INF$), falls es eine IIM M gibt, so daß für alle $L \in \mathcal{L}$ und alle $t \in Text(L)$ [$i \in Info(L)$] gilt:*

(1) *M produziert auf t [i] genau eine Hypothese j und stoppt dann,*

(2) *j ist korrekt für L, i.e. $L = L(G_j)$.*

Eine natürliche Abschwächung von Definition 13 erhält man, indem anstelle keines Hypothesenwechsels eine vorher fixierte Anzahl k von Wechseln als obere Grenze erlaubt wird. Die Anzahl der maximal erlaubten Hypothesenwechsel gibt dann eine obere Schranke für die Komplexität des Lernprozesses an.

Wenn wir uns mit dem Lernen indizierter Familien in Abhängigkeit von der Anzahl der erlaubten Hypothesenwechsel beschäftigen, bietet es sich an, stets folgende technische

Voraussetzung als erfüllt anzunehmen. Seien t ein Text (i ein Informant) und $x \in \mathbb{N}$. Wenn eine IIM M nach dem Lesen von t_x bzw. i_x zum ersten Mal eine Hypothese ausgibt, so soll M nach dem Lesen von t_{x+r} bzw. i_{x+r}, $r \in \mathbb{N}$, stets eine Hypothese auf ihr Ausgabeband schreiben. Es macht nun keine Mühe zu zeigen, daß zu jeder IIM $\hat{M}$ eine IIM M konstruiert werden kann, die der beschriebenen Bedingungung genügt und die bei gleichem Informationsangebot und Hypothesenraum die gleiche Folge von Hypothesenwechseln produziert.

Definition 14. *Seien $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum und $k \in \mathbb{N}$.* ***$\mathcal{L}$ heißt mit höchstens k Hypothesenwechseln bei Textbelehrung*** *[**Informantbelehrung**] **bzgl. $\mathcal{G}$ erlernbar** (Bez. $\mathcal{L} \in LIM_k\text{–}TXT$, $\mathcal{L} \in LIM_k\text{–}INF$), falls es eine IIM M gibt, so daß für alle $L \in \mathcal{L}$ und alle $t \in Text(L)$ [$i \in Info(L)$] gilt:*

(1) $L \in LIM-TXT(M)$ $\quad [L \in LIM-INF(M)]$,

(2) $card(\{x \mid M(t_x) \neq M(t_{x+1})\}) \leq k$ $[card(\{x \mid M(i_x) \neq M(i_{x+1})\}) \leq k]$.

Aus den Definitionen 13 und 14 ergibt sich $FIN\text{–}TXT = LIM_0\text{–}TXT$ und $FIN\text{–}INF = LIM_0\text{–}INF$ unmittelbar.

Mit der folgenden Definition wollen wir Definition 12 dahingehend verschärfen, daß eine IIM nur dann ihre Hypothese wechseln darf, wenn sie einen Widerspruch ihrer aktuellen Hypothese zu den bislang gelesenen Daten entdeckt. Diese Forderung erscheint sinnvoll, da es auf den ersten Blick schwer fällt, sich einen anderen Grund als Dateninkonsistenz vorzustellen, der eine IIM veranlassen könnte, ihre Meinung zu ändern. Angluin (1980a) nannte derartige IIMs ***konservativ*** und gab folgende Definition.

Definition 15. *Seien $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum.* ***$\mathcal{L}$ heißt konservativ bei Textbelehrung*** *[**Informantbelehrung**] **bzgl. $\mathcal{G}$ erlernbar** (Bez. $\mathcal{L} \in CONSERVATIVE\text{–}TXT$, $\mathcal{L} \in CONSERVATIVE\text{–}INF$), falls es eine IIM M gibt, so daß für alle $L \in \mathcal{L}$, alle $t \in Text(L)$ [$i \in Info(L)$] und alle $x \in \mathbb{N}$ gilt:*

(1) $L \in LIM-TXT(M)$ $\quad [L \in LIM-INF(M)]$

(2) *Ist $M(t_x) \neq M(t_{x+k})$ für ein $k \geq 1$, so muß $t^+_{x+k} \not\subseteq L(G_{j_x})$ gelten.* [*Ist $M(i_x) \neq M(i_{x+k})$ für ein $k \geq 1$, so muß $i^+_{x+k} \not\subseteq L(G_{j_x})$ oder $i^-_{x+k} \not\subseteq co\text{–}L(G_{j_x})$ gelten.*]

Konservativ lernende IIM dürfen insbesondere also niemals eine *übergeneralisierte* Hypothese für die zu lernende Sprache ausgeben. Dabei nennen wir eine Hypothese j ***übergeneralisiert*** für L, falls $L \subset L(G_j)$ ist. Hat man nur positive Daten zur Verfügung, so ist das Erkennen übergeneralisierter Hypothesen sicherlich schwer. Wir untersuchen das

Problem, inwieweit nichtkonservatives Lernverhalten überhaupt notwendig ist, ausführlich im Abschnitt 4.1.

Bevor wir zur formalen Definition der verschiedenen Monotoniebegriffe kommen, erklären wir noch ***konsistentes*** Lernen. Konsistent arbeitende IIMs wurden von Barzdin (1974a) sowie Blum und Blum (1975) eingeführt und untersucht. Konsistenz bedeutet, daß eine IIM in jedem Lernschritt den Teil der Informationen richtig widerzuspiegeln hat, in den sie schon Einsicht nehmen konnte.

Definition 16. *Seien $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum.* ***$\mathcal{L}$ heißt konsistent bei Textbelehrung*** *[****Informantbelehrung****]* ***bzgl. $\mathcal{G}$ erlernbar*** *(Bez. $\mathcal{L} \in CONS{-}TXT$, $\mathcal{L} \in CONS{-}INF$), falls es eine IIM M derart gibt, daß für alle $L \in \mathcal{L}$, alle $t \in Text(L)$ $[i \in Info(L)]$ und alle $x \in \mathbb{N}$ gilt:*

(1) $\mathcal{L} \in LIM{-}TXT$ $[\mathcal{L} \in LIM{-}INF]$

(2) *Wenn M auf t_x (i_x) eine Hypothese j_x ausgibt, so ist $t_x^+ \subseteq L(G_{j_x})$ $[i_x^+ \subseteq L(G_{j_x})$ und $i_x^- \subseteq co{-}L(G_{j_x})]$.*

Abschließend in diesem Abschnitt formalisieren wir die in der Einleitung beschriebenen Arten des monotonen und dual monotonen Lernens. Der Einfluß von Monotonieforderungen wurde erstmals von Jantke (1991a, 1991b) und Wiehagen (1991, 1992) im Bereich des Lernens von rekursiven Funktionen und von Patternsprachen studiert. Wir haben die dort gegebenen Definitionen in geeigneter Weise modifiziert, wobei die Monotonieforderungen dem oben beschriebenen Lernziel anzupassen waren.

Definition 17. *Seien $\mathcal{L}$ eine indizierte Familie und $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum.* ***$\mathcal{L}$ heißt***

(A) ***streng-monoton***

(B) ***monoton***

(C) ***schwach-monoton***

bei Textbelehrung *[****Informantbelehrung****]* ***bzgl. $\mathcal{G}$ erlernbar****, falls es eine IIM M gibt, so daß für alle $L \in \mathcal{L}$, alle $t \in Text(L)$ $[i \in Info(L)]$ und alle $x \in \mathbb{N}$ gilt: M lernt $\mathcal{L}$ im Limes bei Text- bzw. Informantbelehrung, und für beliebige aufeinander folgende Hypothesen j_x, j_{x+k} (produziert auf t_x und t_{x+k} bzw. i_x and i_{x+k} für ein $k \geq 1, k \in N$) gilt:*

(A) $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$

(B) $L(G_{j_x}) \cap L \subseteq L(G_{j_{x+k}}) \cap L$

(C) *wenn* $t_{x+k} \subseteq L(G_{j_x})$, *so ist* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$ [*wenn* $i^+_{x+k} \subseteq L(G_{j_x})$ *und* $i^-_{x+k} \subseteq co{-}L(G_{j_x})$, *so ist* $L(G_{j_x}) \subseteq L(G_{j_{x+k}})$].

Die resultierenden Lerntypen bezeichnen wir mit $SMON-TXT$, $SMON-INF$ (streng-monotones Lernen), $MON-TXT$, $MON-INF$ (monotones Lernen) und $WMON-TXT$ sowie $WMON-INF$ (schwach-monotones Lernen).

Beim schwach-monotonen Lernen darf eine IIM also entweder ihre aktuelle Hypothese generalisieren oder, sofern die aktuelle Hypothese nicht konsistent mit den bereits erhaltenen Daten ist, eine *beliebige* neue Hypothese produzieren. Die duale Version dieser Forderung erhählt man, wenn im letzten Satz „generalisieren“ durch „spezialisieren“ ersetzt wird. Formal ergibt sich folgende Definition, die ertstmals in Kapur (1992) betrachtet wurde.

Definition 18. *Seien* $\mathcal{L}$ *eine indizierte Familie und* $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ *ein Hypothesenraum.* ***$\mathcal{L}$ heißt***

(A) ***dual streng-monoton***

(B) ***dual monoton***

(C) ***dual schwach-monoton***

bei Textbelehrung [***Informantbelehrung***] ***bzgl. $\mathcal{G}$ erlernbar***, *falls es eine IIM* M *gibt, so daß für alle* $L \in \mathcal{L}$, *alle* $t \in Text(L)$ [$i \in Info(L)$] *und alle* $x \in \mathbb{N}$ *gilt:* M *lernt* $\mathcal{L}$ *im Limes bei Text- bzw. Informantbelehrung, und für beliebige aufeinander folgende Hypothesen* j_x, j_{x+k} *(produziert auf* t_x *und* t_{x+k} *bzw.* i_x *and* i_{x+k} *für ein* $k \geq 1, k \in N$ *gilt:*

(A) $co{-}L(G_{j_x}) \subseteq co{-}L(G_{j_{x+k}})$

(B) $co{-}L(G_{j_x}) \cap co{-}L \subseteq co{-}L(G_{j_{x+k}}) \cap co{-}L$

(C) *wenn* $t_{x+k} \subseteq L(G_{j_x})$, *so ist* $co{-}L(G_{j_x}) \subseteq co{-}L(G_{j_{x+k}})$ [*wenn* $i^+_{x+k} \subseteq L(G_{j_x})$ *und* $i^-_{x+k} \subseteq co{-}L(G_{j_x})$, *so ist* $co{-}L(G_{j_x}) \subseteq co{-}L(G_{j_{x+k}})$].

Mit $SMON^d-TXT$, $SMON^d-INF$ (dual streng-monotones Lernen), MON^d-TXT, MON^d–INF (dual monotones Lernen) und $WMON^d$–TXT sowie $WMON^d$–INF (dual schwach-monotones Lernen) bezeichnen wir die resultierenden Lerntypen.

Manchmal werden wir auch einige der oben eingeführten Forderungen kombinieren, z.B. strenge Monotonie und Konsistenz. Die daraus resultierenden Lerntypen bezeichnen

wir dann z.B. mit $SMON-CONS-TXT$ oder $SMON-CONS-INF$. Wir drücken damit die Forderung aus, daß die IIM, die den jeweiligen Lernprozeß realisieren soll, sowohl streng-monoton als auch konsistent bei Text- bzw. Informantbelehrung arbeitet.

Die hier vorgestellten Lerntypen werden wir ausfürlich in Kapitel 4 studieren. Wir erinnern an dieser Stelle nochmals daran, daß wir dann ggf. die oben vereinbarten Präfixe E und K verwenden werden, um die Forderungen nach exakter bzw. klassenumfassender Lernbarkeit zu bezeichnen.

Wie bereits oben angekündigt soll an dieser Stelle noch kurz begründet werden, warum wir beim Lernen von Sprachen auf die Forderung, in jedem Lernschritt eine Hypothese zu produzieren, verzichtet haben. Der Grund ist ganz allgemein darin zu sehen, daß die Bedingung, nach jedem Lesen eines weiteren Eingabeworts eine Hypothese zu produzieren, die Lernfähigkeit unter Umständen deutlich einschränken kann. Dieses Phänomen wurde bereits von Angluin (1980a) beschrieben. Zur Illustration sei hier noch ein kleines Beispiel erlaubt. Wir setzen $L_{2j} = \{a, b\}$, $L_{2j+1} = \{a, c\}$ sowie $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$ und fragen, ob $\mathcal{L}$ bei Informantbelehrung streng-monoton bzgl. eines klassentreuen Hypothesenraumes erlernbar wäre, wenn in jedem Schritt eine Hypothese auszugeben ist. Die Antwort ist nein. Zum Beweis genügt es zu analysieren, welche Hypothese von einer IIM M nach dem Lesen von $(a, +)$ ausgegeben werden könnte. Ist diese Hypothese eine Nummer für $\{a, b\}$, so kann $(a, +)$ mit $(c, +)$ und $(w, -)$ für alle $w \in \Sigma^* \setminus \{a, c\}$ zu einem Informanten i für $\{a, c\}$ fortgesetzt werden. Da M die Familie $\mathcal{L}$ bei Informantbelehrung erlernen soll, müßte sich M irgendwann auf eine korrrekte Hypothese für $\{a, c\}$ stabilisieren. Wegen $\{a, b\} \not\subseteq \{a, c\}$ folgt daraus jedoch sofort, daß M mindestens einmal nicht streng-monoton arbeitet. Ist M's erste Hypothese eine Nummer für $\{a, c\}$, so setzt man $(a, +)$ zu einem Informanten für $\{a, b\}$ fort und erhält den gleichen Widerspruch.

Wie unser Beispiel zeigt, wären also selbst triviale Klassen bei Informantbelehrung nicht streng-monoton lernbar, wenn in jedem Lernschritt eine Hypothese zu produzieren ist. Damit wären die monotonen Lerntypen natürlich uninteressant. Durch den Verzicht auf diese Forderung wird nun nicht nur die obige Familie $\mathcal{L}$ selbst bei Textbelehrung trivial inferierbar, sondern, wie wir in Kapitel 4 zeigen werden, verfügen monoton lernende IIMs über ein beachtliches Lernpotential.

3. Zuverlässiges Lernen von Funktionen

In diesem Kapitel beschäftigen wir uns ausführlich mit der prinzipiellen Leistungsfähigkeit zuverlässigen Lernens. Es sei an dieser Stelle nochmal darauf hingewiesen, daß sämtliche Resultate in diesem Kapitel gemeinsam mit E. B. Kinber erzielt wurden (cf., Kinber und Zeugmann, 1985, 1991).

Wir beginnen mit dem Nachweis der in der Einleitung erwähnten Abgeschlossenheitseigenschaften.

3.1. Abgeschlossenheitseigenschaften

In diesem Abschnitt zeigen wir, daß alle oben definierten Typen zuverlässigen Lernens die von Minicozzi (1976) entdeckten Abgeschlossenheitseigenschaften besitzen. Die Abgeschlossenheit gegenüber rekursiv aufzählbaren Vereinigungen und endlicher Varianz von erlernbaren Funktionenklassen verdient aus folgenden Gründen Beachtung. Einerseits sind diese Eigenschaften nicht trivial, da, wie schon oben erwähnt, Klassen $U_1, U_2 \in EX$ existieren, deren Vereinigung $U_1 \cup U_2$ nicht einmal verhaltenskorrekt lernbar ist (cf., Barzdin, 1974b). Andererseits bietet die Abgeschlossenheit gegenüber Vereinigung und endlicher Varianz natürlich die Möglichkeit zur Synthese immer mächtigerer Lernverfahren aus bereits bekannten einfacheren Lernalgorithmen. Damit läßt sich ein wichtiger Aspekt des Lernens wie gelernt wird erfassen. Es sei noch darauf hingewiesen, daß in Aspitis et al. (1992) sehr interessante Resulte vorgestellt wurden, die Ausasgen darüber erlauben, unter welchen Bedingungen für drei oder mehr Klassen aus EX auch deren Vereinigung in EX liegt. Der folgende Satz liefert die Abgeschlossenheit aller zuverlässigen Lerntypen hinsichtlich rekursiv aufzählbarer Vereinigung.

Satz 3.1. *Sei $\mathcal{M} \subseteq \mathcal{TF}$ und möge $ID \in \{REX^a,\ RBC^a,\ REX_{freq}(\frac{1}{n}),\ RBC_{freq}(\frac{1}{n}) \mid n \geq 1,\ n \in \mathbb{N},\ a \in \mathbb{N} \cup \{*\}\}$ sein. Weiterhin sei $(M_i)_{i\in\mathbb{N}}$ eine rekursive Aufzählung von induktiven Inferenzmaschinen, die uniform im Sinne von $\mathcal{M} - ID$ arbeiten. Dann kann aus den Maschinen M_i eine IIM M derart konstruiert werden, daß $\mathcal{M} - ID(M) = \bigcup_{i\in\mathbb{N}} \mathcal{M} - ID(M_i)$ ist.*

Beweis: Ist $ID = REX^a$, so kann die Behauptung wie bei Minicozzi (1976) gezeigt werden. Für die restlichen Fälle transformieren wir Minicozzis (1976) Beweistechnik. Wir weisen noch darauf hin, daß im Fall $ID = REX^a$ der Begriff „Fehlermeldung“ stets durch „Hypothesenwechsel“ zu ersetzen ist. Seien also $f \in \mathcal{M}$ und $(M_i)_{i\in\mathbb{N}}$ eine rekursive Aufzählung von IIMs, die alle im Sinne von $\mathcal{M} - ID$ arbeiten. Wird die Funktion f von

einer der aufgezählten Maschinen erlernt, etwa von M_i, so gilt $M_i(f^k) = (j_k, 1)$ für fast alle k. Andererseits produziert jede nicht f lernende Maschine M_z unendlich viele Fehlermeldungen, d.h. es gilt $M_z(f^k) = (j_k, 0)$ für unendlich viele k. Die gesuchte IIM M, die $\bigcup_{i \in \mathbb{N}} \mathcal{M} - ID(M_i)$ lernt, kann deshalb wie folgt entworfen werden. M simuliert sukzessive mehr und mehr Berechnungen von aufgezählten Maschinen auf sukzessive wachsenden Anfangsstücken von f. Dabei legt M für jede schon simulierte Maschine einen sogenannten Vertrauenszähler an, mit dem die Anzahl der aufeinanderfolgenden Hypothesen in der simulierten Berechnung gezählt wird, die in der 2. Komponente eine 1 enthalten. Diese Anzahl nennen wir Vertrauensgrad. Solange also eine simulierte Maschine M_i ihren Hypothesen vertraut, wird der Vertrauensgrad inkrementell wachsen. Produziert M_i hingegen eine Fehlermeldung, so wird der aktuelle Vertrauensgrad auf 0 zurückgesetzt. Bei der Eingabe eines Anfangsstücks f^k gibt M dann die Hypothese derjenigen Maschine M_i aus, die von allen simulierten Maschinen $M_0, ..., M_k$ den höchsten Vertrauensgrad erreicht hat. Sollte es mehrere Maschinen mit maximalem Vertrauensgrad geben, so wird die Hypothese der Maschine mit dem kleinsten Index ausgegeben. Außerdem sei noch festgelegt, daß M stets eine Fehlermeldung ausgibt, wenn die bei Eingabe von f^{k-1} favorisierte Maschine im nächsten Simulationsschritt eine Fehlermeldung produziert. Formal erhalten wir damit folgende Definition von M.

$M(f^k) =$ „Berechne parallel

$M_0(f^0),\ M_0(f^1),\ ..., M_0(f^k),$
$M_1(f^0),\ ..., M_1(f^{k-1}),$
–
–
–
$M_{k-1}(f^0),\ M_{k-1}(f^1),$
$M_k(f^0)$

und weise jeder Maschine M_i, $0 \leq i \leq k$ ihren Vertrauensgrad $v(i)$ zu, d.h. die größte Zahl $m \leq k-i$, so daß alle Hypothesen $M_i(f^{k-i-m}),\ M_i(f^{k-i-m+1}),\ ...,\ M_i(f^{k-i})$ uniform eine 1 in der 2. Komponente enthalten. Bestimme $w(k) = min\{i | 0 \leq i \leq k,\ v(i) \geq v(z)$ für alle $z \in \{0, ..., k\}\}$.

Gilt für alle betrachteten Maschinen $v(i) = 0$, so gib $(k, 0)$ aus. Andernfalls teste, ob $w(k) = w(k-1)$. Wenn ja, so gib $M_{w(k)}(f^{k-w(k)})$ aus. Ist $w(k) \neq w(k-1)$, so gib $(k, 0)$ aus.“

Wir haben noch zu zeigen, daß M das Verlangte leistet. Möge dazu $f \in \mathcal{M}$ sein.

Fall 1. $f \in \mathcal{M} - ID(M_j)$ für ein $j \in \mathbb{N}$.
Sei j der kleinste Index für den $f \in \mathcal{M} - ID(M_j)$ ist. Dann existiert ein j_0, so daß $M_j(f^k) = (j_k, 1)$ für alle $k \geq j_0$. Damit wächst der Vertrauensgrad von M_j für alle $k \geq j_0$ inkrementell. Da alle Maschinen M_i mit $i < j$ die Funktion f nicht lernen, produziert jede dieser Maschinen unendlich viele Fehlermeldungen. Somit werden die Zahlen $v(i)$ für alle $i < j$ von Zeit zu Zeit auf Null gesetzt, woraus schließlich $w(k) = j$ für alle hinreichend großen k folgt. Also ist $M(f^k) = M_j(f^{k-j})$ für fast alle k. Damit ist die Behauptung für $ID = RBC^a$ unmittelbar klar. Für die verbleibenden Fälle $ID \in \{REX_{freq}(\frac{1}{n}),\ RBC_{freq}(\frac{1}{n})\}$ ergibt sich die Behauptung aus der Tatsache, daß die endliche Verzögerung, mit der M die Hypothesen von M_j ausgibt, keinen Einfluß auf die Häufigkeit der richtigen Hypothese bzw. Hypothesen im Limes hat.

Fall 2. $f \notin \mathcal{M} - ID(M_j)$ für alle $j \in \mathbb{N}$.
In diesem Fall haben für unendlich viele k alle bei Eingabe von f^k betrachteten Maschinen den Vertrauensgrad 0 oder es gilt $w(k-1) \neq w(k)$. In beiden Fällen gibt M unendlich viele Hypothesen der Form $(k, 0)$ aus, produziert also unendlich viele Fehlermeldungen. Damit arbeitet M zuverlässig auf $\mathcal{M}$.

q.e.d.

Mit dem nächsten Satz etablieren wir die Abgeschlossenheit aller definierten zuverlässigen Lerntypen bzgl. endlicher Varianz. Dazu definieren wir endliche Varianz wie folgt. Sei $U \subseteq \mathcal{R}$; dann bezeichne $[U] = \{g \in \mathcal{R} \mid \exists f\ [f \in U,\ f =_* g]\}$ die Klasse aller Funktionen, die man durch endliche Variation der Funktionen aus U erhält. Zur Illustration sei $U = \{0^\infty\}$. Dann ist $[U] = \{\alpha 0^\infty \mid \alpha \in N^*\}$. Dieses Beispiel veranschaulicht die Bedeutung des nachstehenden Satzes, daß sich aus jedem zuverlässigen Lernalgorithmus für eine Klasse U uniform eine IIM M konstruieren läßt, die $[U]$ zuverlässig lernt, schon eindrucksvoll. Denn im Beispiel ist U selbst natürlich trivial erlernbar.

Satz 3.2. *Mögen $\mathcal{M} \subseteq \mathcal{TF}$ mit $[\mathcal{M}] = \mathcal{M}$ und $ID \in \{REX^a,\ RBC^a,\ REX_{freq}(\frac{1}{n}),\ RBC_{freq}(\frac{1}{n}) \mid n \geq 1,\ n \in \mathbb{N},\ a \in \mathbb{N} \cup \{*\}\}$ sein. Weiterhin sei $U \subseteq \mathcal{R} \cap \mathcal{M}$ und $\hat{M}$ eine IIM mit $U \subseteq \mathcal{M} - ID(\hat{M})$. Dann kann aus $\hat{M}$ uniform eine IIM M konstruiert werden, so daß $[U] \subseteq \mathcal{M} - ID(M)$.*

Beweis: Der Beweis läuft im wesentlichen auf eine geschickte Anwendung von Satz 3.1 hinaus. Wir fixieren dazu eine berechenbare eineindeutige Aufzählung der Menge $\mathbb{N}^*$. Möge nun $U \subseteq \mathcal{R} \cap \mathcal{M}$ und $\hat{M}$ eine IIM mit $U \subseteq \mathcal{M} - ID(\hat{M})$ sein. Wir definieren jetzt für jedes $i \in \mathbb{N}$ eine IIM M_i wie folgt. Seien $g \in \mathcal{M}$ und $k \in \mathbb{N}$. Dann arbeitet M_i bei Eingabe von g^k folgendermaßen. Zunächst wird g^k dekodiert. M_i ersetzt jetzt die

ersten Werte $g(0), ..., g(m_i)$ durch $n_0, ..., n_{m_i}$, wobei $(n_0, ..., n_{m_i})$ gerade das i–te Tupel in der Aufzählung von $\mathbb{N}^*$ ist. Danach simuliert M_i die Berechnung von $\hat{M}$ bei Eingabe von $(n_0, ..., n_k)$, falls $k \leq m$ bzw. von $(n_0, ..., n_{m_i}, g(k-m_i+1), ..., g(k))$, falls $k > m_i$. Möge (j_k, b_k) die dabei von $\hat{M}$ berechnete Hypothese sein. M_i gibt dann (h_k, b_k) aus, wobei h_k eine kanonische Nummer der wie folgt definierten Funktion η ist: $\eta(x) = g(x)$, falls $x \leq m_i$ und $\eta(x) = \varphi_{j_k}(x)$, sonst.

Nach Konstruktion arbeitet M_i zuverlässig auf $\mathcal{M}$, weil $[\mathcal{M}] = \mathcal{M}$. Sei nun $(M_i)_{i \in \mathbb{N}}$ die kanonische rekursive Aufzählung aller M_i. Nach Satz 3.1 genügt es jetzt, $[U] \subseteq \bigcup_{i \in \mathbb{N}} \mathcal{M} - ID(M_i)$ zu zeigen. Möge $g \in [U]$ sein. Nach der Definition von $[U]$ existieren ein Tupel $(n_0, ..., n_{m_i})$ und eine Funktion $f \in U$, so daß $f(x) = n_x$ für alle $x \leq m_i$ und $g(x) = f(x)$ für alle $x > m_i$ ist. Sei nun i die Nummer von $(n_0, ..., n_{m_i})$ in der Aufzählung von $\mathbb{N}^*$. Da $\hat{M}$ die Funktion f im Sinne von $\mathcal{M} - ID$ lernt, ergibt sich aus der Konstruktion von M_i unmittelbar $g \in \mathcal{M} - ID(M_i)$.

Offenbar ist die Konstruktion der M_i uniform in $\hat{M}$. Nach Satz 3.1 ist die gesuchte Maschine M damit uniform in den M_i konstruierbar und somit auch in $\hat{M}$.

q.e.d.

Die obigen Sätze erlauben jetzt unmittelbar nachstehende Folgerung, die einen ersten Einblick in die Leistungsfähigkeit zuverlässigen Lernens gestattet.

Corollar 3. 3.

(1) $EX \setminus \mathcal{R}-RBC_{freq}(\frac{1}{n}) \neq \emptyset$ *für alle* $n \in \mathbb{N}$.

(2) $\mathcal{R}-REX^a \subset EX^a$ *für alle* $a \in \mathbb{N} \cup \{*\}$.

(3) $\mathcal{R}-REX_{freq}(\frac{1}{n}) \subset EX_{freq}(\frac{1}{n})$ *für alle* $n \in \mathbb{N}$.

(4) $\mathcal{R}-RBC_{freq}(\frac{1}{n}) \subset BC_{freq}(\frac{1}{n})$ *für alle* $n \in \mathbb{N}$.

Beweis: Podnieks (1974) Hierarchiesatz für die BC–Häufigkeitserkennung impliziert unmittelbar, daß $\mathcal{R} \notin BC_{freq}(\frac{1}{n})$, für alle $n \in \mathbb{N}$. Somit gilt auch $\mathcal{R} \notin \mathcal{R}-RBC_{freq}(\frac{1}{n})$, für alle $n \in \mathbb{N}$. Sei jetzt $U = \{f \mid f \in \mathcal{R},\ \varphi_{f(0)} = f\}$. Offensichtlich ist $U \in EX^a$ für alle $a \in \mathbb{N} \cup \{*\}$. Angenommen, $U \in \mathcal{R}-RBC_{freq}(\frac{1}{n})$ für ein $n \in \mathbb{N}$. Dann folgt aus Satz 3.2 sofort $[U] \in \mathcal{R}-RBC_{freq}(\frac{1}{n})$. Da $[U] = \mathcal{R}$, ist das jedoch unmöglich. Damit ist (1) gezeigt. (2), (3) und (4) ergeben sich völlig analog, da die einfachen Inklusionen offensichtlich sind.

q.e.d.

Im folgenden Abschnitt untersuchen wir die Leistungsfähigkeit der Lerntypen

$\mathcal{M}-REX^a$ für die aus unserer Sicht besonders interessanten Fälle $\mathcal{M} = \mathcal{TF}$ bzw. $\mathcal{M} = \mathcal{R}$.

3.2. Die Leistungsfähigkeit deterministischer zuverlässiger Lernalgorithmen

Ziel dieses Abschnittes ist es, die Leistungsfähigkeit deterministischer zuverlässiger Lernalgorithmen in Abhängigkeit von der Menge $\mathcal{M}$, auf der zuverlässig gelernt werden soll, und der Anzahl der erlaubten Anomalien detaillierter zu studieren.

Wie bereits oben dargelegt wurde, kann es durchaus sinnvoll sein, anstelle der Synthese eines korrekten Programms für die zu lernende Funktion f das Lernziel dahingehend abzuschwächen, daß nur noch ein Programm inferiert werden muß, welches bei fast allen Eingaben x die Werte von f korrekt berechnet. Von besonderem Interesse ist nun die Frage, welchen Zuwachs an Lernfähigkeit man mit dieser Abschwächung des Lernziels prinzipiell erreichen bzw. nicht erreichen kann. Die von Case und Smith (1983) erzielten Resultate zeigen, daß sowohl im Limes als auch verhaltenskorrekt mehr gelernt werden kann. Die dort verwendeten Beweistechniken sind jedoch nicht auf zuverlässige Lernalgorithmen anwendbar. Wir wollen deshalb zuerst die Frage beantworten, ob Anomalien im Falle zuverlässigen Lernens überhaupt zu einer Vergößerung der Lernfähigkeit führen. Wie der folgende Satz zeigt, wird die Lernfähigkeit tatsächlich erheblich erweitert. Zum besseren Verständnis der Schärfe von Aussage (2) des folgenden Satzes merken wir noch an, daß man $EX^a \subseteq EX_{team}(a+1)$ mit einer geeigneten Modifikation der in Daley (1983) vorgestellten Techniken zeigen kann. Ein Team von $a+1$ Maschinen ist also stets in der Lage, a Fehler im Lernprozeß zu korrigieren. Da andererseits schon $EX_{team}(2) \setminus BC \neq \emptyset$ gilt, sehen wir, daß sogar schärfer $EX^a \subset EX_{team}(a+1)$ für alle $a \in \mathbb{N}$ ist. Ein EX–Team von a Maschinen kann im allgemeinen jedoch nicht a Fehler korrigieren, und zwar sogar dann nicht, wenn die entsprechende Funktionenklasse zuverlässig auf der Menge der totalen Funktionen erlernbar ist.

Satz 3.4. *Sei $a > 0$. Dann gilt:*

(1) $\mathcal{TF}-REX^a \setminus EX^{a-1} \neq \emptyset$,

(2) $\mathcal{TF}-REX^a \setminus EX_{team}(a) \neq \emptyset$

(3) $\mathcal{TF}-REX^* \setminus \bigcup_{a\in\mathbb{N}} EX^a \neq \emptyset$

Beweis: Wir zeigen die Behauptungen durch Diagonalisierung. Dazu ist es nötig, die schon von Gold (1965) entwickelte grundlegende Diagonaliserungstechnik entscheidend zu

modifizieren. Wir führen den Beweis deshalb in drei Etappen durch. In ersten Teil zeigen wir Aussage (1). Danach geben wir an, wie die gerade entwickelte Technik zu modifizieren ist, um (3) zu erhalten. Im letzten Teil des Beweises wird (2) gezeigt, da hierzu nochmals eine wesentliche Verfeinerung der Beweistechnik erforderlich wird.

Sei $a > 0$ beliebig fixiert. Wir interpretieren jede partiell-rekursive Funktion als IIM. Nach Gold (1965) gibt es eine Funktion $g \in \mathcal{R}$ derart, daß $EX^a(\varphi_i) \subseteq EX^a(\varphi_{g(i)})$ und $\varphi_{g(i)} \in \mathcal{R}$, für alle $i \in \mathbb{N}$. Zur Vereinfachung der Notation setzen wir $M_i =_{df} \varphi_{g(i)}$. Wir konstruieren jetzt zunächst zu jeder IIM M_i effektiv eine partiell-rekursive Funktion f_i derart, daß f_i zumindest eine rekursive Fortsetzung f besitzt (evtl. f_i selbst), die nicht von M_i im Sinne von EX^{a-1} gelernt wird. Zweckmäßigerweise identifizieren wir dabei rekursive Funktionen mit der Folge ihrer Werte.

Sei i beliebig fixiert. Wir setzen $f_i(0) = i$ und berechnen $n_{i_1} = M_i(i)$. Die Werte $f_i(a+1),\ f_i(a+2), ...$ werden jetzt gleich 0 gesetzt, solange bis ein k_1 derart gefunden wird, daß

(α) $M_i(i\underbrace{02...z}_{\substack{\text{die ersten } a \\ \text{geraden Zahlen}}}0^{k_1}) \neq n_{i_1}$, wobei $z = 2(a-1)$, oder

(β) $M_i(i\underbrace{13...z'}_{\substack{\text{die ersten } a \\ \text{ungeraden Zahlen}}}0^{k_1}) \neq n_{i_1}$, wobei $z' = 2a-1$.

Tritt weder (α) noch (β) ein, so ist f_i schon mit Ausnahme der a Argumente $x = 1, 2, ..., a$ überall definiert. Andererseits wird mindestens eine der Funktionen $i024...z0^\infty$ bzw. $i13...z'0^\infty$ von M_i nicht im Sinne von EX^{a-1} gelernt.

Wenn (α) oder (β) eintritt, so definieren wir f_i wie folgt:
Ist zuerst (α) eingetreten, dann setzen wir $f_i(1) = 0,\ f_i(2) = 2,\ ...\ f_i(a) = 2(a-1)$.
Tritt dagegen (β) zuerst ein, so wird $f_i(1) = 1,\ f_i(2) = 3,\ ...\ f_i(a) = 2a-1$ gesetzt.

In beiden Fällen setzten wir die Definition von f_i wie folgt fort. Berechne $n_{i_2} = M_i(f_i^{a+1+k_1})$. Setze $f_i(k_1+2a+2),\ f_i(k_1+2a+3), ...$ gleich 0, solange bis ein k_2 derart gefunden wird, daß

(α) $M_i(if_i(1)...f_i(a)0^{k_1}02...z0^{k_2}) \neq n_{i_2}$, wobei $z = 2(a-1)$, oder

(β) $M_i(if_i(1)...f_i(a)0^{k_1}13...z'0^{k_2}) \neq n_{i_2}$, wobei $z' = 2a-1$.

Wenn weder (α) noch (β) eintritt, so ist f_i schon mit Ausnahme der a Argumente $x = k_1+a+2,\ ....\ , k_1+2a+2$ überall definiert. Aber mindestens eine der Funktionen

$if_i(1)...f_i(a)0^{k_1}02...z0^\infty$ bzw. $if_i(1)...f_i(a)0^{k_1}13...z'0^\infty$ wird von M_i nicht im Sinne von EX^{a-1} inferiert.

Tritt (α) oder (β) ein, so wird f_i folgendermaßen weiter definiert:
Wenn zuerst (α) eingetreten ist, dann setzen wir $f_i(k_1+a+2)=0$ $f_i(k_1+a+3)=2$, ... $f_i(k_1+2a+2)=2(a-1)$.
Ist zuerst (β) eingetreten, so wird $f_i(k_1+a+2)=1$ $f_i(k_1+a+3)=3$ ... $f_i(k_1+2a+2)=2a-1$ gesetzt.

Durch Iteration der gerade beschriebenen Konstruktion erhalten wir effektiv einen Algorithmus zur Berechnung von f_i aus i. Damit ist $f_i \in \mathcal{P}$. Treten die jeweils relevanten (α) oder (β) unendlich oft ein, so ist f_i sogar überall definiert. Andererseits führt dann M_i bei sukzessiver Eingabe des Graphen von f_i unendlich viele Hypothesenwechsel durch, woraus sofort $f_i \notin EX^{a-1}(M_i)$ folgt. Wir setzten jetzt

$$U = \{f \mid f \in \mathcal{R},\ \exists f_i\ [f_i \subseteq f]\}.$$

Dann gilt nach dem oben Gesagten $U \notin EX^{a-1}$.

Zur Vervollständigung des Beweises von (1) haben wir jetzt noch $U \in \mathcal{TF}-REX^a$ zu zeigen. Dazu definieren wir einen IIM M wie folgt. Sei $t \in \mathcal{TF}$ und $x \in \mathbb{N}$.

$M(t^x)$ = „Starte die Berechnung von $f_{t(0)}(z)$ für alle Argumente $z \leq x$, wobei $f_{t(0)}$ wie in der oben beschriebenen Konstruktion definiert ist. Wenn $f_{t(0)}(z)$ für alle mit Ausnahme von a Argumenten $z \leq x$ berechnet wurde, stoppe die weitere Berechnung von $f_{t(0)}(z)$, $z \in \{0,1,...x\}$ und teste, ob $t(z) = f_{t(0)}(z)$ für alle berechneten Werte von $f_{t(0)}$ ist.

Wenn ja, so gib die kanonische Nummer von $f_{t(0)}$ aus. Andernfalls gib x aus.“

Wir haben noch zu zeigen, daß M das Verlangte leistet. Nach Konstruktion ist für jeden Wert von $t(0)$ eine Funktion $f_{t(0)}$ berechenbar, die für alle mit Ausnahme von höchstens a Argumenten definiert ist. Damit ist zunächst M selbst rekursiv. Konvergiert M auf einer Funktion $t \in \mathcal{TF}$, so hat M die Gültigkeit von $t =_a f_{t(0)}$ im Limes verifiziert. Folglich wird t im Sinne von $\mathcal{TF}-REX^a$ gelernt. Ist dagegen $t \neq_a f_{t(0)}$, so wechselt M ihre Hypothesen unendlich oft. Damit arbeitet M zuverlässig auf der Menge $\mathcal{TF}$. Somit ist (1) bewiesen.

Zum Beweis von (3) nutzt man die gerade eben beschriebene Konstruktion zur Definition von Funktionen f_i, wobei die einzige Modifikation darin besteht, $a = i$ zu setzen.

Damit erhalten wir dann Funktionen f_i, die mindestens eine totale und rekursive Fortsetzung f mit $f \notin EX^i(M_i)$ besitzen. Wird also

$$U = \{f \mid f \in \mathcal{R},\ \exists f_i\ [f_i \subseteq f]\}$$

gesetzt, so erhalten wir unmittelbar $U \notin \bigcup_{a \in \mathbb{N}} EX^a$.

Wir geben jetzt eine IIM M an, die U im Sinne von $\mathcal{TF}\text{-}REX^*$ lernt. Sei $t \in \mathcal{TF}$ und $x \in \mathbb{N}$.

$M(t^x) =$ „Starte die Berechnung von $f_{t(0)}(z)$ für alle Argumente $z \leq x$, wobei $f_{t(0)}$ wie in der oben beschriebenen Konstruktion definiert ist. Wenn $f_{t(0)}(z)$ für alle mit Ausnahme von $t(0)+1$ Argumenten $z \leq x$ berechnet wurde, stoppe die weitere Berechnung von $f_{t(0)}(z)$, $z \in \{0, 1, ...x\}$ und teste, ob $t(z) = f_{t(0)}(z)$ für alle berechneten Werte von $f_{t(0)}$ ist.

Wenn ja, so gib kanonische Nummer von $f_{t(0)}$ aus. Andernfalls gib x aus.“

Es macht nun keine Mühe, mit den gleichen Argumenten wie oben $U \subseteq \mathcal{TF}\text{-}REX^*(M)$ zu zeigen.

Der Beweis von (2) wird hier nur für $a = 2$ vorgeführt, da die weitere Generalisierung der verwendeten Technik dann keine weitere Mühe macht. Wir interpretieren jetzt jede Zahl i als Kodierung eines geordneten Paares (n, m). Konzeptionell verfahren wir wie oben, d.h. wir geben wieder einen Algorithmus zur Berechnung einer Funktion f_i an, die mindestens eine totale Fortsetzung f mit $f \notin EX_{team}(M_n, M_m)$ besitzt. Die technischen Schwierigkeiten des Beweises resultieren aus der Tatsache, daß jetzt die beiden Teammitglieder M_n und M_m ad absurdum geführt werden müssen. Wir erläutern im Verlauf der Definition von f_i, welche Konsequenzen sich aus dieser Zielstellung ergeben.

Sei also $i \in \mathbb{N}$. Dekodiere i, d.h. berechne n und m.

Main LOOP

Wir definieren $f_i(0) = i$ und setzen $f_i(2), f_i(3), ...$ gleich Null, solange bis ein k_1 mit

(α) $M_n(i) \neq M_n(i10^{k_1})$ oder

(β) $M_n(i) \neq M_n(i00^{k_1})$ oder

(γ) $M_m(i) \neq M_m(i10^{k_1})$ oder

(δ) $M_m(i) \neq M_m(i00^{k_1})$

gefunden wird.

Wenn ein derartiges k_1 nicht gefunden wird, so ist f_i bereits vollständig mit Ausnahme des Argumentes 1 definiert. Tritt dagegen (α) oder β ein, so setzen wir $f_i(1) = 1$ bzw. $f_i(1) = 0$. Außerdem wird $q = m$ und $r = n$ gesetzt. Wenn hingegen (γ) oder (δ) zuerst verifiziert wird, so definieren wir $f_i(1) = 1$ bzw. $f_i(1) = 0$; setzen jedoch $q = n$ und $r = m$.

Im Parameter r merken wir uns, welche Maschine des Teams gerade einen Hypothesenwechsel ausgeführt hat. Um nun das Team ad absurdum zu führen, sind wir natürlich besonders daran interessiert, daß nun auch die Maschine M_q einen Hypothesenwechsel macht. Aber M_q muß sich natürlich nicht nach unseren Wünschen richten. Deshalb bauen wir eine zweite Falle ein, die wir **LOOP** nennen.

LOOP

Setze $\tau = f_i^{k_1+1}$. Definiere $f_i(k_1+4),\ f_i(k_1+5),\ \ldots$ als Null, solange bis ein k_2 derart gefunden wurde, daß entweder

(α) $\Phi_{M_q(\tau)}(k_1+2) \leq k_2$*, d.h. $\varphi_{M_q(\tau)}(k_1+2)$ als definiert nachgewiesen werden konnte oder

(β_0) $M_q(\tau) \neq M_q(\tau 0^{k_2+2})$ oder

(β_1) $M_q(\tau) \neq M_q(\tau 10^{k_2+1})$ oder

(γ_{00}) $M_r(\tau) \neq M_r(\tau 0^{k_2+2})$ oder

(γ_{01}) $M_r(\tau) \neq M_r(\tau 010^{k_2})$ oder

(γ_{10}) $M_r(\tau) \neq M_r(\tau 100^{k_2})$ oder

(γ_{11}) $M_r(\tau) \neq M_r(\tau 110^{k_2})$

verifiziert werden konnte.

Wenn keiner der genannten Fälle eintritt, so ist f_i mit Ausnahme der Argumente $x = k_1+2$ und $x = k_1+3$ überall definiert. Sollte (α) zuerst eintreten, so setzten wir $f_i(k_1+2) = 1 \dot{-} \varphi_{M_q}(\tau)(k_1+2)$ und $f_i(k_1+3) = 0$. Damit ist gesichert, daß die aktuelle Hypothese von M_q nicht korrekt sein kann. Wir vertauschen dann die Werte von q und r und iterieren die Konstruktion, indem wir **Main LOOP** mit den neuen Werten starten.

*$\Phi_0, \Phi_1, \ldots$ bezeichnet ein beliebig fixiertes Komplexitätsmaß

Tritt (β_0) oder (β_1) zuerst ein, so haben wir den gewünschten Hypothesenwechsel. Wir setzen dann $f_i(k_1+2)=0$ bzw. $f_i(k_1+2)=1$, vertauschen die Werte von q und r und iterieren die Konstruktion durch Abarbeitung von **Main LOOP**. Möge jetzt einer der (γ_{cd}) Fälle eingetreten sein, wobei $c,d \in \{0,1\}$. Stoppe dann alle Prozeduren. Setze $f_i(k_1+3)=d$ und $z=c$.

Wir weisen darauf hin, daß $f_i(k_1+2)$ noch *nicht* definiert ist. Die aktuelle Situation stellt sich jetzt so dar, daß die Maschine M_r, die wir bereits im letzten Schritt zu einem Hypothesenwechsel zwingen konnten, erneut ihre Hypothese wechselte, wenn wir $f_i(k_1+2)=c$ setzten. Wir halten uns diese Option offen, verstärken jedoch die Anstrengungen, um auch die Maschine M_q hereinzulegen. Deshalb starten wir jetzt **LOOP1**.

LOOP1

Setze wie folgt fort. Definiere $f_i(k_1+k_2+6)$, $f_i(k_1+k_2+7)$, ... gleich Null, solange bis ein k_3 derart gefunden wurde, daß entweder

(A) $\Phi_{M_q(\tau)}(k_1+2) \leq k_2+k_3$, d.h. $\varphi_{M_q(\tau)}(k_1+2)$ als definiert nachgewiesen werden konnte oder

(B_0) $M_r(\tau z f_i(k_1+3)0^{k_2}) \neq M_r(\tau z f_i(k_1+3)0^{k_2}0^{k_3+1})$ oder

(B_1) $M_r(\tau z f_i(k_1+3)0^{k_2}) \neq M_r(\tau z f_i(k_1+3)0^{k_2}10^{k_3})$ oder

(C_0) $M_q(\tau) \neq M_q(\tau z f_i(k_1+3)0^{k_2}0^{k_3+1})$ oder

(C_1) $M_q(\tau) \neq M_q(\tau z f_i(k_1+3)0^{k_2}10^{k_3})$

eintritt.

Mit den Tests in (B_0) und (B_1) suchen wir den nächsten Hypothesenwechsel der Maschine M_r. In (C_0) und (C_1) testen wir, ob die alten Bedingungen (β_0) bzw. (β_1) eintreten, wobei jetzt zu berücksichtigen war, daß $f_i(k_1+3)$ inzwischen definiert ist. Den Test (α) haben wir durch (A) ersetzt, wodurch sich die Komplexitätsschranke für $\Phi_{M_q(\tau)}(k_1+2)$ vergrößert hat.

Tritt jetzt (A) zuerst ein, so setzen wir $f_i(k_1+2)=1\dot{-}\varphi_{M_q}(\tau)(k_1+2)$ und $f_i(k_1+k_2+5)=0$, vertauschen die Werte von q und r und starten wieder **Main LOOP** mit den aktuellen Werten.

Wenn (C_0) oder (C_1) zuerst eintreten, so setzten wir $f_i(k_1+2)=z$ bzw. $f_i(k_1+2)=1\dot{-}z$ und $f_i(k_1+k_2+5)=0$. Anschließend werden die Werte von q und r vertauscht und wir beginnen wieder den **Main LOOP**.

Sollten nun (B_0) oder (B_1) zuerst verifiziert werden, so wird nur $f_i(k_1 + k_2 + 5) = 0$ bzw. $f_i(k_1 + k_2 + 5) = 1$ definiert und **LOOP1** erneut abgearbeitet.

Wir diskutieren jetzt, welche Varianten von f_i auftreten können. Wenn beide Maschinen M_n und M_m nur endlich viele Hypothesenwechsel ausführen und für das relevante k der Wert $\varphi_{M_q(f_i^k)}(k+1)$ nicht definiert ist, so ist f_i eine mit Ausnahme von zwei Stellen überall definierte Funktion. Darüber hinaus sind fast alle Hypothesen von M_q gleich $M_q(f_i^k)$. Somit kann M_q auf der relevanten totalen Fortsetzung von f_i nicht korrekt arbeiten, da $\varphi_{M_q(f_i^k)}(k+1)$ nicht definiert ist. Da nun weiterhin auch die Maschine M_r in (B_0) bzw. (B_1) gezwungen wird, einen Hypothesenwechsel auszuführen, dieser aber nicht auftritt, gibt es auch eine totale Fortsetzung von f_i, auf der M_r versagt. Somit ist das Team in diesem Fall ad absurdum geführt.

Angenommen, nur M_r führt unendlich viele Hypothesenwechsel aus und der Wert $\varphi_{M_q(f_i^k)}(k+1)$ ist nicht definiert. Dann ist f_i überall mit Ausnahme an der Stelle $k+1$ definiert. Sei $f(x) = f_i(x)$ für $x \neq k+1$ und $f(k+1) = z$, wobei z wie in der obigen Konstruktion definiert ist. Dann versagt das Team $M_n,\ M_m$ offensichtlich erneut.

Schließlich sei angenommen, daß die Werte von q und r unendlich oft getauscht werden. Dann ist $f_i \in \mathcal{R}$, aber jedes Teammitglied versagt. Wenn wir also $U = \{f \mid f \in \mathcal{R}\ , \exists f_i[f_i \subseteq f]\}$ setzen, so erhalten wir sofort $U \notin EX_{team}(2)$.

Die gesuchte IIM M, die U im Sinne von $\mathcal{TF}\text{-}REX^2$ inferiert, definieren wir wie folgt. Sei $t \in \mathcal{TF}$ und $x \in \mathbb{N}$.

$M(t^x) =$ „Starte die Berechnung von $f_{t(0)}(z)$ für alle Argumente $z \leq x$, wobei $f_{t(0)}$ wie in der oben beschriebenen Konstruktion definiert ist. Wenn $f_{t(0)}(z)$ für alle mit Ausnahme von 2 Argumenten $z \leq x$ berechnet wurde, stoppe die weitere Berechnung von $f_{t(0)}(z)$, $z \in \{0, 1, ...x\}$ und teste, ob $t(z) = f_{t(0)}(z)$ für alle berechneten Werte von $f_{t(0)}$ ist.

Wenn ja, so gib kanonische Nummer von $f_{t(0)}$ aus. Andernfalls gib x aus.“

Der Nachweis $U \subseteq \mathcal{TF}\text{-}REX^2(M)$ erfolgt dann anolog wie oben.

q.e.d.

Wir wollen jetzt untersuchen, welcher Zuwachs an Lernfähigkeity prinzipiell *nicht* durch erlaubte Anomalien zu erreichen ist. Satz 3.4 könnte zu der Vermutung führen, daß sogar $\mathcal{R}\text{-}REX \subseteq \mathcal{TF}\text{-}REX^*$ gilt, d.h. daß man durch ein abgeschwächtes Lernziel stets die Zuverlässigkeit einer IIM auf einer wesentlich größeren Menge, nämlich auf $\mathcal{TF}$ anstelle von $\mathcal{R}$, erreichen kann.

Wie wir sogleich beweisen werden, ist diese Vermutung falsch. Damit rückt der folgende Satz gleichzeitig die Strenge der Forderung, auf allen totalen Funktionen zuverlässig lernen zu müssen, in die richtige Perspektive. Intuitiv kann man dieses Phänomen folgendermaßen begründen. Eine auf der Menge $\mathcal{TF}$ zuverlässig lernende IIM M hat insbesondere auf jeder Funktion $f \in \mathcal{TF} \setminus \mathcal{R}$ unendlich viele Hypothesenwechsel auszuführen. Da M andererseits auf jeder Funktion aus der zu lernende Klasse U konvergieren muß, ist intuitiv klar, daß es zumindest für fast alle Hypothesenwechsel einen für M erkennbaren Grund geben muß. Die Forderung nach der Berechenbarkeit von M impliziert weiterhin, daß M die Notwendigkeit fast aller Hypothesenwechsel irgendwie auf berechenbare Art und Weise zu ermitteln hat. Da nun die totalen, jedoch nicht rekursiven Funktionen gerade dadurch charakterisiert sind, daß sie sich in ihrer Gesamtheit jeder Form der Berechenbarkeit entziehen, bleibt als Alternative nur die Möglichkeit, aus den Graphen der totalen Funktionen Vermutungen zu berechnen, die im weiteren Verlauf des Lernprozesses gegebenenfalls falsifizierbar sind. An dieser Stelle kommen aber die unterschiedlichen topologischen Eigenschaften der Mengen $\mathcal{TF}$ und $\mathcal{R}$ ins Spiel. Bezüglich der Baireschen Metrik d ($d(f,g) = 1/(\mu x[f(x) \neq g(x)])$[*] für alle $f, g \in \mathcal{TF}$) sind nämlich die Mengen $\mathcal{TF}_X$ genau dann kompakt, wenn X beschränkt ist, während die Mengen $\mathcal{R}_X$ diese Eigenschaft nicht besitzen (cf., Grabowski, 1986). Induktive Inferenzmaschinen können nun in geeigneter Art und Weise als stetige Abbildungen interpretiert werden (cf., Osherson et al., 1986). Stetige Abbildungen haben aber auf kompakten Mengen andere Eigenschaften als auf nicht kompakten Mengen. So müssen sie beispielsweise auf kompakten Mengen ihr Minumum respektive Maximum annnehmen. Aus dieser Einsicht heraus erklärt sich, warum auf $\mathcal{R}$ zuverlässig lernbare Klassen von Funktionen mit uniform beschränktem Wertebereich topologisch komplizierter strukturiert sein können als Klassen, die auf der Menge $\mathcal{TF}$ zuverlässig erlernbar sind. Diesen Sachverhalt werden wir auch im Abschnitt 3.4 noch mehrfach ausnutzen.

Wir wenden die eben gewonnene Einsicht jetzt an, um $\mathcal{R}\text{-}REX \setminus \mathcal{TF}\text{-}REX^* \neq \emptyset$ zu beweisen. Dazu bot es sich an, die Lerntypen $\mathcal{R}\text{-}REX^a$ und $\mathcal{TF}\text{-}REX^a$ zunächst in komplexitätstheoretischen Termini zu charakterisieren. Der folgende Satz besagt, daß die zuverlässige Lernbarkeit einer Klasse U äquivalent dazu ist, vermöge eines berechenbaren Operators aus den Graphen der zu lernenden Funktionen eine rekursive Majorante für deren Berechnungskomplexität an fast allen Stellen zu berechnen. Die Unterschiede zwischen zuverlässiger Lernbarkeit auf der Menge der totalen bzw. rekursiven Funktionen werden dabei durch den geforderten Definitionsbereich der berechenbaren Operatoren wi-

[*] μ bezeichnet den μ–Operator

dergespiegelt. Zur formalen Definition von berechenbaren Operatoren verweisen wir auf Rogers (1967). Weiterführende Informationen zu berechenbaren Operatoren finden sich z.B. in Helm (1971) und Zeugmann (1984). Sei $\mathcal{O}$ ein Operator, der Funktionen in Funktionen abbildet. Dann bezeichne $\mathcal{O}(f,x)$ den Funktionswert der Funktion $\mathcal{O}(f)$ an der Stelle x. Mögen jetzt $\mathcal{O}$ ein beliebig fixierter Operator und $M, Z \subseteq \mathcal{PF}$ sein. Dann verwenden wir die Schreibweise „$\mathcal{O}(M) \subseteq Z$“ anstelle von „$M \subseteq Arg\ \mathcal{O}$, und aus $f \in M$ folgt $\mathcal{O}(f) \in Z$“. Schließlich sei noch vereinbart, einen berechenbaren Operator $\mathcal{O}$ *monoton* zu nennen, wenn für alle Funktionen $f,\ g \in Arg\ \mathcal{O}$ gilt, daß aus $f(x) \leq g(x)$ für fast alle x die Ungleichung $\mathcal{O}(f,x) \leq \mathcal{O}(g,x)$ für fast alle x folgt.

Satz 3.5. *Sei $a \in \mathbb{N} \cup \{*\}$. Dann gilt für jede Klasse $U \subseteq \mathcal{R}$:*
$U \in \mathcal{TF}\text{-}REX^a$ ($U \in \mathcal{R}\text{-}REX^a$) genau dann, wenn es einen berechenbaren Operator $\mathcal{O}$ mit $\mathcal{O}(\mathcal{TF}) \subseteq \mathcal{TF}_a$ ($\mathcal{O}(\mathcal{R}) \subseteq \mathcal{R}_a$) derart gibt, daß zu jeder Funktion $f \in U$ ein Programm j mit $\varphi_j =_a f$ und $\Phi_j(x) \leq \mathcal{O}(f,x)$ für fast alle x existiert.

Beweis: Notwendigkeit. Sei $U \subseteq \mathcal{TF}\text{-}REX^a(M)$ ($U \subseteq \mathcal{R}\text{-}REX^a(M)$) für eine IIM M. Den gewünschten Operator $\mathcal{O}$ definieren wir wie folgt. Sei $f \in \mathcal{TF}$ ($f \in \mathcal{R}$) und $x \in \mathbb{N}$.

$\mathcal{O}(f,x) =$ „Berechne $M(f^x)$. Nutze jeweils die halbe Zeit, um (1) und (2) auszuführen, bis (3) oder (4) eintritt.

(1) Berechne $M(f^{x+1}),\ M(f^{x+2}), \ldots$

(2) Teste, ob $\Phi_{M(f^x)}(x) = y$ für $y = 0,\ 1,\ 2,\ \ldots$

(3) In (1) wird ein $k \in \mathbb{N}$ derart gefunden, daß $M(f^x) \neq M(f^{x+k})$. Setze $\mathcal{O}(f,x) = 0$.

(4) In (2) wird ein $y \in \mathbb{N}$ mit $\Phi_{M(f^x)}(x) = y$ gefunden. Setze dann $\mathcal{O}(f,x) = \Phi_{M(f^x)}(x)$.“

Offenbar ist der Operator $\mathcal{O}$ berechenbar. Wir haben noch zu zeigen, daß er die im Satz genannten Bedingungen erfüllt.

***Hilfssatz* 1.** $\mathcal{O}(\mathcal{TF}) \subseteq \mathcal{TF}_a$ ($\mathcal{O}(\mathcal{R}) \subseteq \mathcal{R}_a$).

Sei $f \in \mathcal{TF}$ ($f \in \mathcal{R}$) und $x \in \mathbb{N}$. Angenommen, $\mathcal{O}(f,x)$ ist nicht definiert. Wir haben zu zeigen, daß dieser Fall höchstens a mal eintreten kann. Da $\mathcal{O}(f,x)$ nicht definiert ist, kann insbesondere (3) nicht eingetreten sein. Somit muß $M(f^x)$ schon diejenige Hypothese sein, gegen die M auf f konvergiert. Da M zuverlässig auf $\mathcal{TF}$ (auf $\mathcal{R}$) arbeitet, gilt

deshalb $\varphi_{M(f^x)} =_a f$. Also ist $\Phi_{M(f^x)} \in \mathcal{R}_a$. Folglich kann (4), sofern (3) nicht eintritt, höchstens a mal nicht eintreten. Damit ist $\mathcal{O}(\mathcal{TF}) \subseteq \mathcal{TF}_a$ ($\mathcal{O}(\mathcal{R}) \subseteq \mathcal{R}_a$).

q.e.d. (Hilfssatz 1)

***Hilfssatz* 2**. Sei $f \in U$. Dann existiert ein j mit $\varphi_j =_a f$ und $\Phi_j(x) \leq \mathcal{O}(f,x)$ für fast alle x.

Da $f \in U$ muß f von der IIM M mit höchstens a Anomalien gelernt werden. Somit existiert $j = \lim_{x\to\infty} M(f^x)$ und es gilt $\varphi_j =_a f$. Damit kann aber in der Berechnung von $\mathcal{O}(f,x)$ Bedingung (3) nur endlich oft eintreten. Folglich ist $\mathcal{O}(f,x) = \Phi_j(x)$ für fast alle $x \in \mathbb{N}$.

q.e.d. (Hilfssatz 2)

Hinlänglichkeit. Wir definieren die gesuchte IIM als limes-rekursives Funktional wie folgt. Sei $f \in \mathcal{TF}$ ($f \in \mathcal{R}$).

$M(f) =$ „Führe Anweisung 0 aus:
Anweisung n: Interpretiere n als Kodenummer eines geordneten Paares natürlicher Zahlen (j,k) und berechne diese. Gib j aus.
Teste für alle $x \in \mathbb{N}$ für die sich $\mathcal{O}(f,x)$ als definiert erwiesen hat, ob $\Phi_j(x) \leq max\{k, \mathcal{O}(f,x)\}$ und $\varphi_j(x) = f(x)$.
Wird ein $x \in \mathbb{N}$ mit $\mathcal{O}(f,x)$ definiert aber $\Phi_j(x) > max\{k, \mathcal{O}(f,x)\}$ oder $\Phi_j(x) \leq max\{k, \mathcal{O}(f,x)\}$ und $\varphi_j(x) \neq f(x)$ gefunden, so beende die Abarbeitung von Anweisung n und beginne Anweisung $n+1$ abzuarbeiten."

Es macht nun keine Mühe zu zeigen, daß M zuverlässig auf $\mathcal{TF}$ bzw. $\mathcal{R}$ arbeitet.

Ist weiterhin $f \in U$, so existieren nach Voraussetzung Zahlen j und k mit $\varphi_j =_a f$ und $\Phi_j(x) \leq max\{k, \mathcal{O}(f,x)\}$ für alle x für die $\mathcal{O}(f,x)$ definiert ist. Sei n die Kodenummer von (j,k). Dann kann M die Abarbeitung der Anweisung n offenbar nie beenden. Folglich hat M auf f zu konvergieren. Da M zuverlässig auf $\mathcal{TF}$ bzw. $\mathcal{R}$ arbeitet, wird damit f mit höchstens a Anomalien gelernt.

q.e.d.

Wir merken noch an, daß sich die unterschiedliche topologische Struktur von auf $\mathcal{TF}$ bzw. $\mathcal{R}$ zuverlässig erlernbaren Funktionenklassen U jetzt nach Satz 3.5 durch komplexitätstheoretische Eigenschaften der von den entsprechenden Operatoren generierten Operatorkomplexitätsklassen

$$\mathcal{R}_{\mathcal{O}} =_{df} \{f \in \mathcal{TF} \mid \exists j\ [\varphi_j =_a f,\ \Phi_j(x) \leq \mathcal{O}(f,x)]\}$$

beschreiben läßt. Damit haben wir die Möglichkeit geschaffen, den folgenden Satz zumindest konzeptionell sehr übersichtlich zu beweisen.

Satz 3.6. $\mathcal{R}-REX \setminus \mathcal{TF}-REX^* \neq \emptyset$

Beweis: Wir erläutern zunächst die Struktur des Beweises. Mit den nachstehenden Lemmata zeigen wir folgende komplexitätstheoretische Eigenschaft aller im Sinne $\mathcal{TF}-REX^*$ erlernbaren Klassen rekursiver Prädikate. Wenn $U \subseteq \mathcal{R}_{\{0,1\}}$ und $U \in \mathcal{TF}-REX^*$, so existiert stets eine Funktion $t \in \mathcal{R}$ derart, daß es zu jedem Prädikat $f \in U$ ein Programm j mit $\varphi_j = f$ und $\Phi_j(x) \leq t(x)$ für unendliche viele $x \in \mathbb{N}$ gibt. Das heißt die Komplexität aller durch eine IIM M lernbaren Prädikate kann zumindest an unendlich vielen Stellen uniform durch eine rekursive Funktion t beschränkt werden.

Im zweiten Teil des Beweises arbeiten wir dann heraus, daß es Klassen rekursiver Prädikate gibt, die zuverlässig auf der Menge $\mathcal{R}$ gelernt werden können und die der eben genannten komplexitätstheoretischen Bedingung nicht genügen. Diese Klassen sind somit nicht im Sinne von $\mathcal{TF}-REX^*$ inferierbar, womit der gewünschte Beweis erbracht ist.

***Lemma* A**. *Zu jedem berechenbaren Operator $\mathcal{O}$ mit $\mathcal{O}(\mathcal{TF}) \subseteq \mathcal{TF}_*$ kann ein berechenbarer Operator $\hat{\mathcal{O}}$ mit $\hat{\mathcal{O}}(\mathcal{TF}) \subseteq \mathcal{TF}$ derart konstruiert werden, daß für alle $t \in \mathcal{TF}$ und unendlich viele $x \in \mathbb{N}$ gilt $\mathcal{O}(t,x) = \hat{\mathcal{O}}(t,x)$.*

Beweis: Wir definieren den gewünschten Operator $\hat{\mathcal{O}}$ wie folgt. Sei $t \in \mathcal{TF}$.

$\hat{\mathcal{O}}(t,0) =$ „Suche ein $x \in \mathbb{N}$ für das sich $\mathcal{O}(t,x)$ als definiert erweist.
Wenn das erste derartige x gefunden wurde, setzte $\hat{\mathcal{O}}(t,0) = ... = \hat{\mathcal{O}}(t,x) =_{df} \mathcal{O}(t,x)$."

Möge jetzt x die zuletzt gefundene Zahl sein, für die sich $\mathcal{O}(t,x)$ als definiert erwies. Setze dann wie folgt fort.

$\hat{\mathcal{O}}(t,x+1) =$ „Suche ein $x' > x$ für das sich $\mathcal{O}(t,x')$ als definiert erweist.
Wenn das erste derartige x' gefunden wurde, setzte $\hat{\mathcal{O}}(t,x+1) = ...\hat{\mathcal{O}}(t,x') =_{df} \mathcal{O}(t,x')$."

Aus der Definition des Operators $\hat{\mathcal{O}}$ ergibt sich unmittelbar seine Berechenbarkeit, da der Operator $\mathcal{O}$ berechenbar und $\mathcal{O}(t,x)$ für fast alle $x \in \mathbb{N}$ definiert ist. Folglich wird stets ein x' mit $\mathcal{O}(t,x')$ definiert gefunden. Damit gilt $\hat{\mathcal{O}}(\mathcal{TF}) \subseteq \mathcal{TF}$ und nach Konstruktion weiterhin $\mathcal{O}(t,x) = \hat{\mathcal{O}}(t,x)$ für unendlich viele x.

q.e.d. (Lemma A)

***Lemma* B. (Meyer und Fischer, 1968)**

Zu jedem berechenbaren Operator $\mathcal{O}$ mit $\mathcal{O}(\mathcal{TF}) \subseteq \mathcal{TF}$ existiert ein berechenbarer und monotoner Operator $\tilde{\mathcal{O}}$ derart, daß $\tilde{\mathcal{O}}(\mathcal{TF}) \subseteq \mathcal{TF}$ und $\mathcal{O}(t,x) \leq \tilde{\mathcal{O}}(t,x)$ für alle $t \in \mathcal{TF}$ und fast alle $x \in \mathbb{N}$.

Mit dem folgenden Lemma erhalten wir die oben annoncierte komplexitätstheoretische Eigenschaft von im Sinne $\mathcal{TF}-REX^*$ erlernbaren Klassen rekursiver Prädikate. Dazu benötigen wir noch eine wichtige Eigenschaft von Komplexitätsmaßen. Wir wollen ein Komplexitätsmaß $\Phi_0,\ \Phi_1,\ \Phi_2, \ldots$ ***endlich invariant*** nennen, wenn für alle Funktionen $f,\ g \in \mathcal{P}$ mit $f =_* g$ und alle Funktionen $t \in \mathcal{R}$ gilt: es existiert genau dann ein j mit $\varphi_j = f$ und $\Phi_j \leq t(x)$ für fast alle $x \in Arg\ f$, wenn es ein k mit $\varphi_k = g$ und $\Phi_k \leq t(x)$ für fast alle $x \in Arg\ g$ gibt. Wie Lewis (1971) gezeigt hat, sind viele natürliche Komplexitätsmaße endlich invariant.

Lemma **C**. *Sei $U \subseteq \mathcal{R}_{\{0,1\}}$ mit $U \in \mathcal{TF}-REX^*$. Dann gibt es eine Schranke $t \in \mathcal{R}$, so daß zu jeder Funktion $f \in U$ ein Programm j mit $\varphi_j = f$ und $\Phi_j(x) \leq t(x)$ für unendlich viele $x \in \mathbb{N}$ existiert.*

Beweis: Da der Lerntyp $\mathcal{TF}-REX^*$ nicht von der Wahl des Komplexitätsmaßes abhängt, sei im folgenden $\Phi_0,\ \Phi_1,\ \Phi_2, \ldots$ endlich invariant. Sei $U \subseteq \mathcal{R}_{\{0,1\}}$ mit $U \in \mathcal{TF}-REX^*$. Dann existiert nach Satz 3.5 ein berechenbarer Operator $\mathcal{O}$ mit $\mathcal{O}(\mathcal{TF}) \subseteq \mathcal{TF}_*$ und $U \subseteq \mathcal{R}_{\mathcal{O}}$. Zu diesem Operator kann nach Lemma A ein Operator $\hat{\mathcal{O}}$ mit $\hat{\mathcal{O}}(\mathcal{TF}) \subseteq \mathcal{TF}$ und $\mathcal{O}(t,x) = \hat{\mathcal{O}}(t,x)$ für unendlich viele x konstruiert werden. Zum Operator $\hat{\mathcal{O}}$ existiert nach Lemma B ein monotoner Operator $\tilde{\mathcal{O}}$, der den Operator $\hat{\mathcal{O}}$ beschränkt. Wir setzen $t =_{df} \tilde{\mathcal{O}}(1^\infty)$. Da nun $U \subseteq \mathcal{R}_{\mathcal{O}}$ ist, existiert zunächst für jedes $f \in U$ ein Programm k mit $\varphi_k =_* f$ und $\Phi_k(x) \leq \mathcal{O}(f,x)$ für fast alle x. Da das Komplexitätsmaß endlich invariant ist, gibt es somit auch ein Programm j mit $\varphi_j = f$ und $\Phi_j(x) \leq \mathcal{O}(f,x)$ für fast alle x. Damit erhalten wir folgende Ungleichung:

$$\Phi_j(x) \leq \mathcal{O}(f,x) \leq \hat{\mathcal{O}}(f,x) \leq \tilde{\mathcal{O}}(f,x) \leq t(x) \text{ für unendlich viele } x,$$

wobei die letzte Relation aus der Monotonie von $\tilde{\mathcal{O}}$ und $U \subseteq \mathcal{R}_{\{0,1\}}$ folgt.

q.e.d. (Lemma C)

Wir beenden den Beweis mit folgendem Lemma.

Lemma **D**. *Es gibt eine Klasse $U \subseteq \mathcal{R}_{\{0,1\}}$ mit*

(1) $U \in \mathcal{R}-REX$,

(2) *für jede Funktion $t \in \mathcal{R}$ existiert ein $f \in U$, so daß für alle Programme j von f die Bedingung $\Phi_j(x) > t(x)$ für fast alle x erfüllt ist.*

Beweis: Lemma D ist bereits in Blum und Blum (1975) annonciert. Für einen vollständigen Beweis verweisen wir auf Kinber und Zeugmann (1985).

q.e.d. (Lemma D)

Sei jetzt U eine nach Lemma D gewählte Klasse. Dann muß $U \notin \mathcal{TF}-REX^*$ gelten, da sich andernfalls ein Widerspruch zu Lemma C ergäbe. Also ist $U \in \mathcal{R}-REX \setminus \mathcal{TF}-REX^*$.

q.e.d.

Der letzte Satz erlaubt unmittelbar nachstehendes Corollar.

Corollar 3. 7.

(1) $\mathcal{TF}-REX^a \subset \mathcal{R}-REX^a$ *für alle* $a \in \mathbb{N} \cup \{*\}$.

(2) $\mathcal{TF}-REX^{a+1} \# \mathcal{R}-REX^a$ *für alle* $a \in \mathbb{N}$.

Zusammenfassend ergeben die Sätze 3.4 und 3.6 sowie Corollar 3.7 hinsichtlich der Leistungsfähigkeit deterministischer zuverlässiger Lernalgorithmen folgendes Bild.

Deterministische zuverlässige Inferenz versus Lernen im Limes

$$\begin{array}{ccccccccc}
\mathcal{TF}-REX & \subset & \mathcal{TF}-REX^1 & \subset & \ldots & \subset & \bigcup_{a\in\mathbb{N}} \mathcal{TF}-REX^a & \subset & \mathcal{TF}-REX^* \\
\cap & & \cap & & & & \cap & & \cap \\
\mathcal{R}-REX & \subset & \mathcal{R}-REX^1 & \subset & \ldots & \subset & \bigcup_{a\in\mathbb{N}} \mathcal{R}-REX^a & \subset & \mathcal{R}-REX^* \\
\cap & & \cap & & & & \cap & & \cap \\
EX & \subset & EX^1 & \subset & \ldots & \subset & \bigcup_{a\in\mathbb{N}} EX^a & \subset & EX^*
\end{array}$$

Abbildung 1

3.3. Monte-Carlo Inferenz und zuverlässige Häufigkeitserkennung

Wir wenden uns jetzt dem Vergleich von Monte-Carlo Inferenz und zuverlässiger Häufigkeitserkennung zu. Dabei werden wir Pitts (1989a) Unifikationsresultate in den Bereich des zuverlässigen Lernens übertragen. Als unmittelbare Konsequenz liefern uns die nachstehenden Sätze gleichzeitig interessante Einsichten in eventuelle Hierarchien zuverlässiger Häufigkeitserkennung. Wie wir sehen werden, muß jede dieser Hierarchien diskret sein. Es sei noch angemerkt, daß man unter Verwendung der von Podnieks (1975) entwickelten Beweistechniken kürzere direkte Beweise für die Diskretheit eventuell vorhandener Hierarchien der zuverlässigen Häufigkeitserkennung führen kann. Da wir in diesem Abschnitt jedoch umfassendere Ziele verfolgen, bot es sich an, Pitts (1989a) Beweistechniken geeignet zu modifizieren. Damit der Umfang der vorliegender Arbeit nicht über

Gebühr ausgedehnt wird, setzen wir Vertrautheit des Lesers mit Pitts Beweistechniken voraus und beschreiben nur die auszuführenden Modifikationen.

Bevor wir zum ersten Satz dieses Abschnittes kommen, sei noch vereinbart, für eine probabilistische IIM P und $f \in \mathcal{TF}$ mit $\mathcal{B}_{P,f}$ den vollständigen binären Baum zu bezeichnen, der alle möglichen Berechnungsverläufe von P repräsentiert.

Satz 3.8. *Seien $n \in \mathbb{N}$ und $\mathcal{M} \subseteq \mathcal{TF}$. Weiterhin möge $U \in \mathcal{M}-RBC_{prob}(p)$ für ein p mit $1 \geq p > 1/(n+1)$ sein. Dann existiert eine IIM M, die folgenden Bedingungen genügt:*

(1) *M lernt U auf $\mathcal{M}$ BC–zuverlässig mit Häufigkeit $1/n$,*

(2) *für jedes $f \in U$ existiert ein $r \in \{0, ..., n-1\}$ derart, daß die Folge $((j_k, b_k))_{k \in \mathbb{N}}$ der von M auf f produzierten Hypothesen der Eigenschaft $\varphi_{j_k} = f$ für fast alle k mit $k \equiv r \bmod n$ genügt.*

Beweis: Unser Beweis koinzidiert im wesentlichen mit Pitts Beweis von $BC_{prob}(p) \subseteq BC_{team}(n)$ (cf., Pitt, 1989, Theorem 3.8, pp. 402 ff.). Wir beschreiben deshalb nur die auszuführenden Modifikationen, insbesondere also die IIM M, die U auf $\mathcal{M}$ BC–zuverlässig mit Häufigkeit $1/n$ lernt und dabei gleichzeitig Bedingung (2) erfüllt. Dabei besteht die Idee zunächst darin, wieder ein Team von n Maschinen zu entwerfen. Wenn keine dieser n Maschinen die zu lernende Funktion f inferiert, so arbeiten alle Teammitglieder zuverlässig, d.h. sie produzieren uniform unendlich viele Fehlermeldungen. Wird dagegen die betrachtete Funktion f von einer Maschine aus dem Team gelernt, so geben auch die anderen IIMs des Teams nur endlich viele Fehlermeldungen aus, und zwar auch dann, wenn sie f *nicht* lernen. Somit arbeiten dann möglicherweise $n-1$ Teammitglieder *nicht zuverlässig*. Die gewünschte IIM M, die U auf $\mathcal{M}$ BC–zuverlässig mit Häufigkeit $1/n$ lernt gibt einfach die Hypothesen der Teammitglieder in rotierender Reihenfolge aus, womit einerseits die gewünschte Häufigkeit richtiger Hypothesen gesichert und andererseits Eigenschaft (2) erfüllt wird.

Möge also $U \subseteq \mathcal{M}-RBC_{prob}(p)(P)$ für eine probabilistische IIM P und ein p mit $1 \geq p > 1/(n+1)$ sein. Wir sichern zunächst, daß für jede Funktion $f \in \mathcal{M}$, auf der P unendlich viele Fehlermeldungen produziert, auch jedes Teammitglied unendlich viele Fehlermeldungen ausgibt. Wir erinnern noch einmal daran, daß P entweder unendlich viele Fehlermeldungen auf allen möglichen Zufallsfolgen produziert oder auf keiner. Gibt die probabilistische IIM P auf $f \in \mathcal{M}$ nur endlich viele Fehlermeldungen aus, so fällt unsere Definition der Teammitglieder mit Pitts Definition zusammen.

Das wird wie folgt erreicht. Wir führen in Pitts Definition einen zusätzlichen Parameter d ein, der initial den Wert 1 zugewiesen bekommt. Der Berechnungsbaum $\mathcal{B}_{P,f}$ wird jetzt schrittweise konstruiert, d.h. für $f \in \mathcal{M}$ und $k = 1,\ 2,\ 3,\ ....$ berechnet sich jede Maschine des Teams zunächst den endlichen Teilbaum der Höhe k von $\mathcal{B}_{P,f}$. Dann testet jede Maschine, ob jeder Weg von der Wurzel zu einem Blatt mindestens d Fehlermeldungen enthält. Ist dies der Fall, so gibt jede Maschine $(k, 0)$ als Hypothese aus und erhöht den Wert von d inkrementell. Sollte der Test nicht erfüllt sein, so berechnet jedes Teammitglied seine Hypothese genau wie in Pitts Definition der Maschine M_i und gibt diese mit einer 1 in der 2. Komponente versehen aus. Als einzige Modifikation ist ein kanonisches Zeitregime einzuführen, um Anweisung (4) in Pitts Definition von M_i ausführbar zu gestalten. Wenn keine Hypothese in der vorgegebenen Zeit berechnet werden kann, so wird $M(f^k) =_{df} M(f^{k-1})$ gesetzt. Zur Vervollständigung definieren wir noch $M_i(f^0) =_{df} (0,0)$ für alle $i \in \{0, 1, ..., n-1\}$. Die gewünschte IIM M gibt jetzt folgende Hypothesenfolge aus. Die ersten n Hypothesen von M sind gerade die n ersten Hypothesen von $M_0,\ M_1, ...,\ M_{n-1}$; die nächsten n Hypothesen von M sind die n zweiten Hypothesen von $M_0,\ M_1, ...,\ M_{n-1}$, usw.

Aus der Konstruktion ergibt sich unmittelbar, daß M unendlich viele Fehlermeldungen produziert, wenn $f \notin \mathcal{M}\text{-}RBC_{prob}(p)(P)$. Möge P jetzt auf f nur endlich viele Fehlermeldungen ausgeben. Sei d_0 die kleinste Anzahl von Fehlermeldungen entlang eines Weges in $\mathcal{B}_{P,f}$. Somit kann für $d > d_0$ der obige Test nicht erfüllt sein. Nach Definition von $\mathcal{M}\text{-}RBC_{prob}$ wird f also von der Monte-Carlo IIM P verhaltenskorrekt mit Wahrscheinlichkeit p erlernt. Somit verhalten sich alle definierten Maschinen M_i im Limes genau wie Pitts Teammitglieder. Folglich gibt es mindestens eine Maschine, sagen wir M_r, die f verhaltenskorrekt lernt. Nach Definition von M erhalten wir unmittelbar $\varphi_{i_k} = f$ für fast alle k mit $k \equiv r \bmod n$, wenn $((j_k, b_k))_{k \in \mathbb{N}}$ die von M produzierte Folge von Hypothesen ist.

q.e.d.

Mit dem nächsten Satz zeigen wir nun, daß jede Hierarchie für zuverlässige Häufigkeitserkennung diskret sein muß. Der Frage, ob derartige Hierarchien für $\mathcal{M} = \mathcal{TF}$ oder $\mathcal{M} = \mathcal{R}$ wirklich existieren, gehen wir im nächsten Abschnitt ausführlich nach. Hier beweisen wir gleichzeitig noch, daß es stets möglich ist, die Hypothesenfolgen aller gelernten Funktionen in gewisser Weise zu normieren.

Satz 3.9. *Sei $\mathcal{M} \subseteq \mathcal{TF}$, $n \in \mathbb{N}$ und $U \in \mathcal{M}\text{-}RBC_{freq}(p)$ für ein p mit $1 \geq p > 1/(n+1)$. Dann gibt es eine IIM M mit:*

(1) $U \in \mathcal{M}\text{-}RBC_{freq}(\frac{1}{n})$,

(2) *für jedes* $f \in \mathcal{M}-RBC_{freq}(\frac{1}{n})(M)$ *existiert ein* $r \in \{0,...,n-1\}$ *derart, daß die von* M *auf* f *produzierte Hypothesenfolge* $((j_k,b_k))_{k\in\mathbb{N}}$ *der Bedingung* $\varphi_{j_k} = f$ *für fast alle* k *mit* $k \equiv r \bmod n$ *genügt.*

Beweis: Diesen Satz beweist man analog wie Theorem 3.33 in Pitt (1989), wobei folgende kleine Modifikation vorzunehmen ist.

Sei $f \in \mathcal{M}$ und $k \in \mathbb{N}$. Möge weiterhin $U \subseteq \mathcal{M}-RBC_{freq}(p)(M')$ für ein p mit $1 \geq p > 1/(n+1)$ sein. Ist $M'(f^k) = (i_k, 0)$, so gibt jedes Teammitglied M_0, ..., M_{n-1} bei Eingabe von f^k auch $(i_k, 0)$ aus. Ist hingegen $M'(f^k) = (i_k, 1)$, so verhält sich jede Maschine des Teams genau wie in Pitts Team und ergänzt die Hypothesen durch eine 1 in der 2. Komponente. Die gewünschte IIM M, die U im Sinne von $\mathcal{M}-RBC_{freq}(\frac{1}{n})$ lernt, ist dann genau wie oben definiert, d.h. sie gibt die Hypothesen der Teammitglieder wieder in rotierender Reihenfolge aus.

q.e.d.

Der letzte Satz erlaubt insbesondere die Folgerung, daß verhaltenskorrekte Monte-Carlo Inferenz und BC–zuverlässige Häufigkeitserkennung prinzipiell von gleicher Leistungsfähigkeit sind.

Corollar 3. 10. *Sei* $\mathcal{M} \subseteq \mathcal{TF}$. *Dann ist* $\mathcal{M}-RBC_{prob}(\frac{1}{n}) = \mathcal{M}-RBC_{freq}(\frac{1}{n})$ *für jede Zahl* $n \geq 1$.

Beweis: Nach Satz 3.8 gilt $\mathcal{M}-RBC_{prob}(\frac{1}{n}) \subseteq \mathcal{M}-RBC_{freq}(\frac{1}{n})$. Zum Beweis der Inklusion $\mathcal{M}-RBC_{freq}(\frac{1}{n}) \subseteq \mathcal{M}-RBC_{prob}(\frac{1}{n})$ wenden wir zunächst Satz 3.9 an. Sei also $U \subseteq \mathcal{M}-RBC_{freq}(\frac{1}{n})(M)$, für eine IIM M, die insbesondere Bedingung (2) von Satz 3.9 erfüllt. Weiterhin sei $f \in \mathcal{M}$ und $((j_k,b_k))_{k\in\mathbb{N}}$ die von M auf f produzierte Hypothesenfolge. O.B.d.A. können wir annehmen, das die gesuchte probabilistische IIM P mit einer n–seitigen fairen Münze ausgestattet ist. P wirft diese Münze genau einmal und erhält eine Zahl $r \in \{0,...,n-1\}$. P gibt dann nur diejenigen Hypothesen (j_k,b_k) aus der Folge $((j_k,b_k))_{k\in\mathbb{N}}$ aus, für die $k \equiv r \bmod n$ ist.

Wird f nicht von M gelernt, so folgt aus der Konstruktion in Satz 3.9 unmittelbar, daß jede von P produzierte Hypothesenfolge unendlich viele Fehlermeldungen enthält. Möge jetzt $f \in \mathcal{M}-RBC_{freq}(\frac{1}{n})(M)$ sein. Dann liefert der Münzwurf von P mit Wahrscheinlichkeit $1/n$ die Zahl $r \in \{0,...,n-1\}$, für die an den Stellen k mod n fast überall richtige Hypothesen für die gerade zu lernende Funktion stehen. Außerdem folgt aus der Konstruktion von M, daß unabhängig vom Ausgang des Münzwurfes keine der für P möglichen Hypothesenfolgen unendlich viele Fehlermeldungen enthält. Also ist $f \in \mathcal{M}-RBC_{prob}(\frac{1}{n})(P)$.

q.e.d.

Der Rest dieses Abschnittes ist dem Nachweis gewidmet, daß die Sätze 3.8 und 3.9 sowie Corollar 3.10 mutatis mutandis auch im Falle EX–zuverlässiger Häufigkeitserkennung gelten.

Satz 3.11. *Seien $n \in \mathbb{N}$ und $\mathcal{M} \subseteq \mathcal{TF}$. Weiterhin möge $U \in \mathcal{M}-REX_{prob}(p)$ für ein p mit $1 \geq p > 1/(n+1)$ sein. Dann existiert eine IIM M, die folgenden Bedingungen genügt:*

(1) *M lernt U auf $\mathcal{M}$ EX–zuverlässig mit Häufigkeit $1/n$,*

(2) *für jedes $f \in U$ existieren ein Programm j und ein $r \in \{0, ..., n-1\}$ derart, daß die Folge $((j_k, b_k))_{k \in \mathbb{N}}$ der von M auf f produzierten Hypothesen den Eigenschaften $j_k = j$ und $\varphi_j = f$ für fast alle k mit $k \equiv r \bmod n$ genügt.*

Beweis: Der Beweis dieses Satzes wird im wesentlichen wie der Beweis von Theorem 3.21 in Pitt (1989) geführt, der $EX_{prob}(p) \subseteq EX_{team}(n)$ unter den obigen Voraussetzungen liefert. Dabei führen wir mutatis mutandis die gleichen Modifikationen wie im Beweis von Satz 3.8 durch. Die gewünschte IIM M simuliert also wieder Pitts EX–team $M_0, \ ... \ , M_{n-1}$, wobei jede Maschine M_i bei Eingabe von f^k zunächst testet, ob im Teilbaum der Höhe k von $\mathcal{B}_{P,f}$ jeder Weg von der Wurzel zu einem Blatt mindestens d Fehlermeldungen enthält. Wenn ja, so gibt jedes M_i die Hypothese $(k, 0)$ aus, erhöht d inkrementell und setzt mit f^{k+1} fort. Andernfalls arbeitet M_i genau wie Pitts i–tes Teammitglied. Schließlich gibt die gewünschte Maschine M wieder die Hypothesen des Teams in rotierender Ordnung aus.

q.e.d.

Satz 3.12. *Sei $\mathcal{M} \subseteq \mathcal{TF}$, $n \in \mathbb{N}$ und $U \in \mathcal{M}-REX_{freq}(p)$ für ein p mit $1 \geq p > 1/(n+1)$ sein. Dann gibt es eine IIM M mit:*

(1) *$U \in \mathcal{M}-REX_{freq}(\frac{1}{n})(M)$,*

(2) *für jedes $f \in \mathcal{M}-REX_{freq}(\frac{1}{n})(M)$ existieren ein Programm j und ein $r \in \{0, ..., n-1\}$ derart, daß die von M auf f produzierte Hypothesenfolge $((j_k, b_k))_{k \in \mathbb{N}}$ den Bedingungen $j_k = j$ und $\varphi_j = f$ für fast alle k mit $k \equiv r \bmod n$ genügt.*

Beweis: Der Beweis kann völlig analog zu Pitts Beweis von Theorem 3.35 gführt werden, wobei die gleichen Modifikationen wie in Satz 3.9 vorzunehmen sind.

q.e.d.

Die Sätze 3.11 und 3.12 erlauben jetzt unmittelbar das nachstehende Corollar.

Corollar 3. 13. *Sei $\mathcal{M} \subseteq \mathcal{TF}$. Dann ist $\mathcal{M}-REX_{prob}(\frac{1}{n}) = \mathcal{M}-REX_{freq}(\frac{1}{n})$ für jede Zahl $n \geq 1$.*

Damit haben wir Monte-Carlo Inferenz vollständig charakterisiert. Im weiteren Verlauf der Arbeit genügt es somit, zuverlässige Häufigkeitserkennung zu studieren. Das soll im nächsten Abschnitt geschehen.

3.4. Die Leistungsfähigkeit zuverlässiger Häufigkeitserkennung

Das wesentliche Ziel dieses Abschnitts besteht darin, einen grundlegenden Einblick in die Leistungsfähigkeit zuverlässiger Häufigkeitserkennung zu erhalten. Dabei beschränken wir uns auf die Fälle $\mathcal{M} = \mathcal{TF}$ und $\mathcal{M} = \mathcal{R}$, da sie von grundlegendem Interesse sind. Wir werden zunächst einige Beobachtungen vorstellen, die erste Einsichten dazu vermitteln, was mit einer bestimmten Häufigkeit gelernt bzw. prinzipiell nicht gelernt werden kann. Ausgehend von den im letzten Abschnitt erzielten Resultaten genügt es, nur die diskreten Häufigkeiten $1,\ 1/2, ..., 1/n, ...$ zu betrachten. Per definitionem gilt für jedes $n \geq 1$ offensichtlich $\mathcal{TF}-REX_{freq}(\frac{1}{n}) \subseteq \mathcal{TF}-RBC_{freq}(\frac{1}{n}) \subseteq \mathcal{R}-RBC_{freq}(\frac{1}{n})$. Wenn es um die Leistungsfähigkeit zuverlässiger Häufigkeitserkennung geht, werden wir, sofern es angebracht ist, stets die Lerntypen $\mathcal{TF}-REX_{freq}(\frac{1}{n})$ bzw. $\mathcal{TF}-RBC_{freq}(\frac{1}{n})$ betrachten. Sollen hingegen die prinzipiellen Grenzen des zuverlässigen Lernens aufgezeigt werden, so betrachten wir vorrangig die Lerntypen $\mathcal{R}-REX_{freq}(\frac{1}{n})$ sowie $\mathcal{R}-RBC_{freq}(\frac{1}{n})$.

Wie wir bereits mit Corollar 3.3 gezeigt haben, muß auch im Falle der Häufigkeitserkennung die Zuverlässigkeit des Lernens stets mit einer Einbuße an Lernfähigkeit bezahlt werden. Wie der dort gegebene Beweis zeigt, gibt es im Limes erlernbare Klassen rekursiver Funktionen, die für kein $n \in \mathbb{N}$ verhaltenskorrekt mit Häufigkeit $1/n$ inferierbar sind. Andererseits zeigt Satz 3.4, daß schon deterministische zuverlässige Lernalgorithmen über ein beachtliches Lernpotential verfügen. Wir wollen jetzt inhaltlich an diesen Satz anknüpfen und dabei die Beantwortung der Frage vorbereiten, ob der Übergang von syntaktischer zu semantischer Konvergenz auch im Falle der zuverlässigen Häufigkeitserkennung stets zu einem Zuwachs an Lernfähigkeit führt.

Satz 3.14. $\mathcal{TF}-RBC \setminus EX^* \neq \emptyset$

Beweis: Zum Beweis dieses Satzes nutzten wir die von Barzdin (1974b) entwickelte Verfeinerung der Goldschen (1965) Diagonalisierungstechnik mit der er insbesondere $EX^* \subset BC$ zeigen konnte. Dazu interpretieren wir wieder jede partiell-rekursive Funktion als induktive Inferenzmaschine. Unter Verwendung der im Beweis von Satz 3.4 angewen-

deten Goldschen (1965) Idee zeigt man leicht die Existenz einer Funktion $g \in \mathcal{R}$, so daß $EX^*(\varphi_i) \subseteq EX^*(\varphi_{g(i)})$ und $\varphi_{g(i)} \in \mathcal{R}$ für alle $i \in \mathbb{N}$. Wir setzen wieder $M_i =_{df} \varphi_{g(i)}$ für alle $i \in \mathbb{N}$. Wir merken noch an, daß die Konstruktion der Klasse $U \in \mathcal{TF}-RBC \setminus EX^*$ *nicht* effektiv ist. Konzeptionell verfahren wir wie folgt. Zu jedem $i \in \mathbb{N}$ wird zunächst eine Klasse U_i mit $card(U_i) \leq 2$ derart definiert, daß $U_i \notin EX^*(M_i)$. Die gesuchte Klasse U wird dann gerade die Vereinigung aller Klassen U_i sein.

Möge $i \in \mathbb{N}$ sein. Wir setzen $f_i(0) = i$ und berechnen $n_{i_1} = M_i(i)$. Wir setzen die Konstruktion wie folgt fort.

Teste für $k = 1,\ 2,\ 3, ...$ parallel, ob (α) oder (β) eintritt.

(α) $M_i(i1^k) \neq n_{i_1}$,

(β) $M_i(i0^k) \neq n_{i_1}$.

Tritt weder (α) noch (β) ein, so setzen wir $U_i = \{i0^\infty, i1^\infty\}$. Folglich gilt dann $U_i \notin EX^*(M_i)$. Möge nun ein k_1 gefunden werden, so daß (α) oder (β) erfüllt ist.

Ist (α) zuerst eingetreten, so definiere $f_i(x) = 1$ für $1 \leq x \leq k_1$.

Ist (β) zuerst eingetreten, so definiere $f_i(x) = 0$ für $1 \leq x \leq k_1$.

In beiden Fällen setze folgendermaßen fort.
Berechne $n_{i_2} = M_i(f_i^{k_1})$. Teste für $k = 1,\ 2,\ 3, ...$ parallel, ob (α) oder (β) eintritt.

(α) $M_i(f_i(0), ..., f_i(k_1)1^k) \neq n_{i_2}$,

(β) $M_i(f_i(0), ..., f_i(k_1)0^k) \neq n_{i_2}$.

Tritt weder (α) noch (β) ein, so setzen wir $U_i = \{f_i(0)...f_i(k_1)0^\infty, f_i(0)...f_i(k_1)1^\infty\}$. Somit erhalten wir erneut $U_i \notin EX^*(M_i)$. Wurde hingegen ein k_2 das (α) oder (β) genügt, so verfahren wir folgendermaßen.

Ist (α) zuerst eingetreten, so definiere $f_i(x) = 1$ für $k_1 + 1 \leq x \leq k_1 + k_2$.

Ist (β) zuerst eingetreten, so definiere $f_i(x) = 0$ für $k_1 + 1 \leq x \leq k_1 + k_2$.

Wir iterieren die gerade beschriebene Konstruktion. Dann kann noch der Fall eintreten, daß unendlich viele k_j gefunden werden, die die jeweils aktuellen Tests (α) oder (β) erfüllen. In diesem Fall ist $f_i \in \mathcal{R}$ und die Folge $(M_i(f_i^n))_{n \in \mathbb{N}}$ konvergiert nicht. Wir

setzten $U_i = \{f_i\}$ und erhalten erneut $U_i \notin EX^*(M_i)$. Schließlich sei $U = \bigcup_{i \in \mathbb{N}} U_i$. Somit gilt nach Konstruktion $U \notin EX^*$.

Es bleibt die Existenz einer IIM M zu zeigen, die U BC–zuverlässig auf $\mathcal{TF}$ lernt. Möge dazu $f \in \mathcal{TF}$ und $n \in \mathbb{N}$ sein. Wir definieren:

$M(f^n) =$ „Simuliere die Konstruktion von $U_{f(0)}$ bis der die mögliche Struktur von $U_{f(0)}$ repräsentierende Baum die Höhe n erreicht hat. Dieser endliche Baum hat höchstens zwei Zweige, da genau ein Zweig gestrichen wird, sofern eine der relevanten Bedingungen (α) oder (β) eingetreten ist. Ist das Anfangsstück $f(0), ..., f(n)$ von allen vorliegenden Zweigen verschieden, so gib $(n, 0)$ aus. Andernfalls gib $(e, 1)$ aus, wobei e eine kanonisches Programm der folgenden Funktion η ist. Für alle $x \leq n$ sei $\eta(x) = f(x)$. Der Algorithmus zur Berechnung von $\eta(x)$ für $x > n$ arbeitet folgendermaßen:

Setze die Simulation der Konstruktion von $U_{f(0)}$ sukzessive bis zur Höhe x fort. Dabei sind zwei Fälle möglich.

Fall 1. Weder die relevante Bedingung (α) noch (β) tritt für das relevante k_j ein.

Besitzt der $U_{f(0)}$ repräsentierende Baum bis zur Höhe n genau einen Zweig, so muß im Punkt n die vorherige Bedingung (α_{alt}) oder (β_{alt}) eingetreten sein. Wir definieren $\eta(x)$ in diesem Fall deshalb so wie im Unterfall 2.3, den wir unten beschreiben. Andernfalls hat der $U_{f(0)}$ entsprechende Baum bis zur Höhe n genau zwei Zweige. Anhand der M vorliegenden Werte von f kann der Zweig ausgewählt werden, für den die Werte von f mit denen im Zweig dargestellten bis zur Stelle n übereinstimmen. Die weiteren Werte dieses Zweiges repräsentieren dann die Werte von $\eta(x)$ für $x > n$ (cf., Abbildung 2).

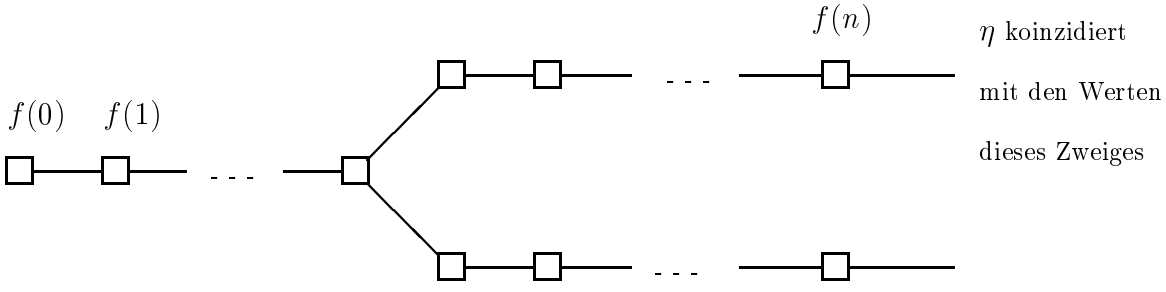

Abbildung 2

Fall 2. Eine der relevante Bedingungen (α) oder (β) ist für das relevante k_j eingetreten.

Dann haben wir zwischen den folgenden Unterfällen zu unterscheiden.

Unterfall 2.1. Der Zweig, der mit $\eta(y)$ für alle $y \leq n$ koinzidiert, repräsentiert die in (α) ausgeführten Berechnungen und (β) tritt ein.

Nach Konstruktion wird dann der Zweig gestrichen, der die in (α) ausgeführten Berechnungen darstellt. Solange die Berechnung in diesem Zweig noch ausgeführt wurde, solange ist $\eta(x)$ als Wert dieses Zweiges im Punkt x definiert. Ist x größer als dieser Punkt, so ist $\eta(x)$ nicht definiert (cf., Abbildung 3).

Unterfall 2.2. Der Zweig, der mit $\eta(y)$ für alle $y \leq n$ koinzidiert, repräsentiert die in (β) ausgeführten Berechnungen und (α) tritt ein.

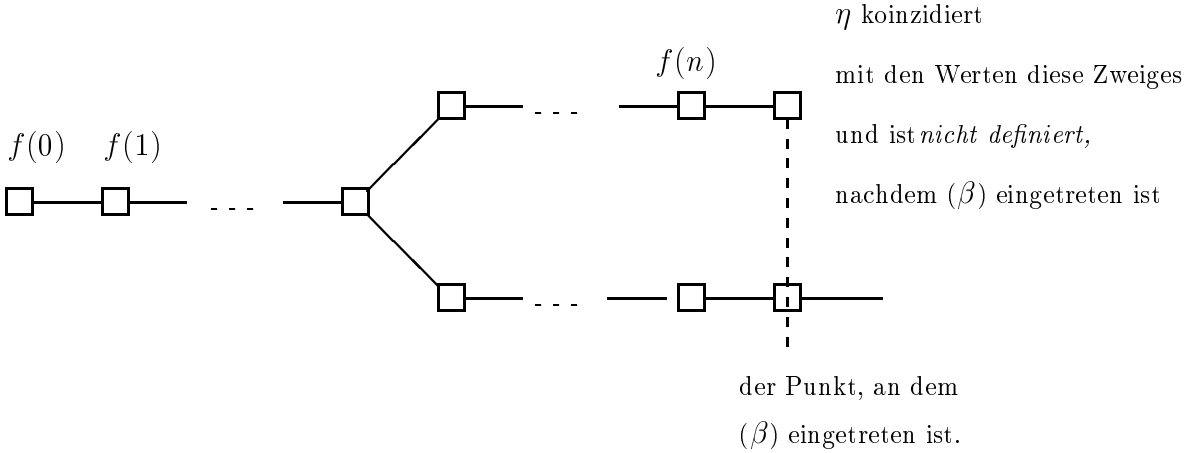

Abbildung 3

Dieser Fall wird analog zu Unterfall 2.1 behandelt.

Unterfall 2.3. Der Zweig, der mit $\eta(y)$ für alle $y \leq n$ koinzidiert, repräsentiert die in (α) ausgeführten Berechnungen und (α) tritt ein.

Nach Konstruktion wird dann der Zweig gelöscht, der die in (β) durchgeführten Berechnungen repräsentiert. Danach wird die Konstruktion iteriert, d.h. (α_{neu}) und (β_{neu}) werden mit neuen Werten gestartet. Ist jetzt x kleiner oder gleich dem Punkt, indem der Zweig für (β) gestrichen wird, so ist $\eta(x)$ als Wert des Punktes x im Zweig für (α) definiert. Ist x größer als der Punkt an dem der Zweig gelöscht wurde, der die in (β) ausgeführten Berechnungen darstellte, so wird der Baum für $U_{f(0)}$ sukzessive weiter simuliert, bis eine der Bedingungen (α_{neu}) oder (β_{neu}) eintritt. Tritt weder (α_{neu}) noch (β_{neu}) ein, so ist $\eta(x)$ nicht definiert. Andernfalls wird eine der Bedingungen (α_{neu}) oder (β_{neu}) zuerst verifiziert, sagen wir im Punkt y und der die andere Bedingung repräsentierende Zweig gestrichen. Ist $x \leq y$, so wird $\eta(x)$ gleich dem im nicht gestrichenen Zweig für x berechneten Wert gesetzt. Sollte $x > y$ sein, so iterieren wir die

Konstruktion, d.h. starten erneut (α) und (β) mit neuen Werten und verfahren wie beschrieben (cf., Abbildung 4).

Unterfall 2.4. Der Zweig, der mit $\eta(y)$ für alle $y \leq n$ koinzidiert, repräsentiert die in (β) ausgeführten Berechnungen und (β) tritt ein.

Dieser Fall kann analog zu Unterfall 2.3 behandelt werden.

Wir haben noch zu zeigen, daß M die Klasse U BC–zuverlässig auf der Menge $\mathcal{TF}$ lernt. Wir unterscheiden zwei Fälle.

Fall 1. $f \notin U$

Nach Konstruktion gilt offensichtlich $f \notin U_{f(0)}$. Somit sind fast alle Anfangsstücke von f vom aktuell simulierten Baum verschieden. Also gibt M für fast alle $n \in \mathbb{N}$ die Hypothese $(n, 0)$ aus.

Fall 2. $f \in U$

Dann wissen wir zunächst $f \in U_{f(0)}$. Möge $card(U_{f(0)}) = 2$ sein. Dann wird $U_{f(0)}$. durch einen Baum repräsentiert, der aus einem endlichen Zweig als Anfangsstück und zwei unendlichen Zweigen besteht, die die relevanten (α) und (β) darstellen. Sei n_0 der Verzweigungspunkt dieses Baumes. Dann gibt M für alle $n > n_0$ nach Konstruktion eine korrekte Hypothese für f aus.

Ist $card(U_{f(0)}) = 1$, so gibt M nur korrekte Hypothesen aus. Damit ist der Satz bewiesen.

q.e.d.

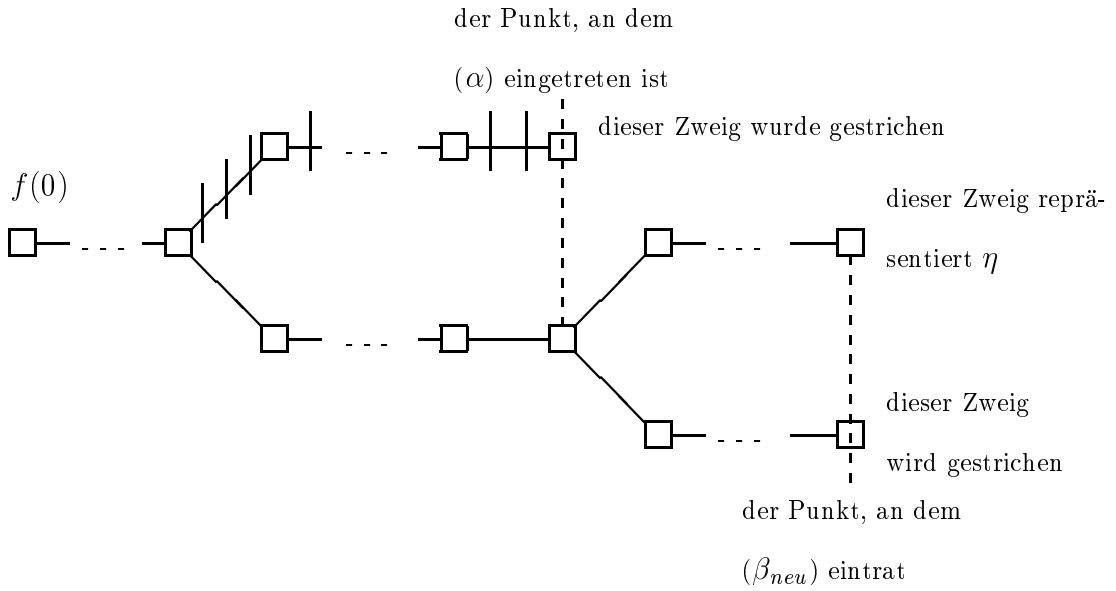

Abbildung 4

In dem folgenden Corollar fassen wir unsere bislang erhaltenen Kenntnisse über BC–zuverlässiges Lernen zusammen.

Corollar 3. 15.

(1) $EX^* \# \mathcal{TF}\text{–}RBC$ *und* $EX^* \# \mathcal{R}\text{–}RBC$

(2) *Sei* $\mathcal{M} \in \{\mathcal{TF}, \mathcal{R}\}$. *Dann gilt:* $\mathcal{M}\text{–}REX^* \subset \mathcal{M}\text{–}RBC$

Beweis: Da $EX \subset EX^*$ und $\mathcal{R}\text{–}RBC = \mathcal{R}\text{–}RBC_{freq}(1)$ ist Aussage (1) unmittelbare Folgerung aus Corollar 3.3 und Satz 3.14.

$\mathcal{M}\text{–}RBC \setminus \mathcal{M}\text{–}REX^* \neq \emptyset$ ist klar nach Satz 3.14. Die einfache Inklusion $\mathcal{M}\text{–}REX^* \subseteq \mathcal{M}\text{–}RBC$ beweist man analog zu $EX^* \subseteq BC$ (cf., Case und Smith, 1983).

q.e.d.

Mit dem nächsten Satz beantworten wir die Frage, ob auch im Falle zuverlässiger Häufigkeitserkennung unter semantischer Konvergenz mehr erlernbar ist als unter syntaktischer.

Satz 3.16. *Sei* $\mathcal{M} \in \{\mathcal{TF}, \mathcal{R}\}$. *Dann gilt:*

$$\mathcal{M}\text{–}REX_{freq}(\tfrac{1}{n}) \subset \mathcal{M}\text{–}RBC_{freq}(\tfrac{1}{n}).$$

Beweis: Die einfache Inklusion ergibt sich unmittelbar aus der Definition der Lerntypen. Zum Nachweis von $\mathcal{M}\text{–}RBC_{freq}(\frac{1}{n}) \setminus \mathcal{M}\text{–}REX_{freq}(\frac{1}{n}) \neq \emptyset$ wenden wir zunächst Satz 3.4, Aussage (2) an. Möge also U eine Klasse mit $U \in \mathcal{M}\text{–}REX^n \setminus EX_{team}(n)$ sein. Dann muß auch $U \in \mathcal{M}\text{–}REX^n \setminus \mathcal{M}\text{–}REX_{freq}(\frac{1}{n})$ gelten, da $EX_{freq}(\frac{1}{n}) = EX_{team}(n)$ (cf., Pitt, 1989) und $\mathcal{M}\text{–}REX_{freq}(\frac{1}{n}) \subset EXfreq(\frac{1}{n})$ nach Corollar 3.3 sowie der offensichtlichen Inklusion $\mathcal{TF}\text{–}REX_{freq}(\frac{1}{n}) \subseteq \mathcal{R}\text{–}REX_{freq}(\frac{1}{n})$. Aus Corollar 3.15 erhalten wir weiterhin $U \in \mathcal{M}\text{–}RBC$. Schließlich gilt $\mathcal{M}\text{–}RBC \subseteq \mathcal{M}\text{–}RBC_{freq}(\frac{1}{n})$ für jede Zahl $n \in \mathbb{N}$, woraus sich jetzt die Behauptung unmittelbar ergibt.

q.e.d.

Daß ein begrenzter Nichtdeterminus im Lernprozeß zu einer Vergrößerung der erlernbaren Familien rekursiver Fuktionen führt, wurde zuerst von Podnieks (1974, 1975) entdeckt. Smith (1982) fand eine weitere Möglichkeit, Pluralismus in den Lernprozeß zu integrieren und klärte dabei die Beziehungen zwischen im Limes und verhaltenskorrekt arbeitenen Teams vollständig. Insbesondere konnte er zeigen, daß für alle $n \in \mathbb{N}$ $EX_{team}(n+1) \setminus BC_{team}(n) \neq \emptyset$ gilt. Wie sich in unseren Untersuchungen herausstellte, sind aber weder Podnieks (1974, 1975) noch Smiths (1982) Beweistechniken auf das zuverlässige Lernen übertragbar. Der wesentliche Grund dafür ist die systematische Ausnutzung der Vereinigungsinstabilität der Lerntypen EX und BC. Wie wir schon im Anschluß

an Definition 10 festgestellt hatten, macht es auch keinen Sinn, Teams von endlich vielen zuverlässigen Maschinen zu betrachten. Pitt (1989a) konnte sogar zeigen, daß selbst ein in Teamform eingeführter unbegrenzter Nichtdeterminismus die Leistungsfähigkeit zuverlässigen Lernens im Limes nicht erweitert. Somit war es zumindest zweifelhaft, ob sich der in der zuverlässigen Häufigkeiterkennung eingeführte begrenzte Pluralismus positiv auf die Leistungsfähigkeit derartiger Lernalgorithmen auswirkt. Ein genaueres Studium des Beweises von Satz 3.1 gestattet die Aussage, daß zuverlässige Häufigkeitserkennung deshalb bezüglich Vereinigung abgeschlossen ist, weil *alle nicht lernbaren Funktionen* uniform zu unendlich vielen Fehlermeldungen führen. Somit ließ sich zumindest eine Zunahme der Leistungskraft von $\mathcal{M}-REX_{freq}(\frac{1}{n})$ vermuten. Interessanterweise gilt nun sogar folgender Satz.

Satz 3.17. *$\mathcal{TF}-REX_{freq}(\frac{1}{n+1}) \setminus BC_{team}(n) \neq \emptyset$ für alle Zahlen $n \geq 1$.*

Beweis: Der folgende Beweis ist technisch kompliziert. Wir werden hier deshalb nur den Fall $n = 1$ ausführlich behandeln. Für die Verallgemeinerung auf beliebige n verweisen wir auf Kinber und Zeugmann (1991). Die prinzipielle Idee ist ist ähnlich zu der im Beweis von Satz 3.14 verwendeten Methode. Wir interpretieren wieder jede partiell-rekursive Funktion als IIM. Barzdin (1974b) zeigte, daß es eine effektive Prozedur $h \in \mathcal{R}$ mit $BC(\varphi_i) \subseteq BC(\varphi_{h(i)})$ und $\varphi_{h(i)} \in \mathcal{R}$ für alle $i \in \mathbb{N}$ gibt. Zur Vereinfachung der Notation sei $M_i =_{df} \varphi_{h(i)}$ für alle $i \in \mathbb{N}$ gesetzt. Wir werden in der Folge *nichteffektiv* zu jeder IIM M_i eine Klasse $U_i \subseteq \mathcal{R}$ mit $card(U_i) = 1$ und $U_i \notin BC(M_i)$ angeben. Dabei gilt entweder $U_i = \{\tau 0^\infty\}$ für ein $\tau \in \mathbb{N}^*$ oder $U_i = \{f\}$ für ein $f \in \mathcal{R}$ mit $f \neq_* 0^\infty$. Schließlich setzen wir $U = \bigcup_{i \in \mathbb{N}} U_i$ und erhalten unmittelbar $U \notin BC$. Bei der Konstruktion der Klassen U_i ist so vorzugehen, daß später $U \in \mathcal{TF}-REX_{freq}(\frac{1}{2})$ beweisbar ist. Die neue Situation gegenüber dem Beweis von Satz 3.14 besteht darin, daß wir jetzt gegen verhaltenskorrekt arbeitende IIMs zu diagonalisieren haben. Hierzu genügt es *nicht* mehr, unenendlich viele Hypothesenwechsel zu erzwingen. Vielmehr sind wir gezwungen, jede Maschine *semantisch* ad absurdum zu führen. Um die Lernbarkeit der zu konstruierenden Klasse U im Sinne von $\mathcal{TF}-REX_{freq}(\frac{1}{2})$ sicher zu stellen, kodieren wir zunächst in den ersten Funktionswert die Nummer i der IIM , die wir zur Definition von U_i verwendet haben. Der Funktionswert $f(1)$ wird so definiert, daß es im Fall $f \neq_* 0^\infty$ unendlich viele Zahlen $y \in \mathbb{N}$ mit $\Phi_{f(1)}(x) \leq f(x+1)$ und $\varphi_{f(1)}(x) = f(x)$ für alle $x \leq y$ gibt und diese Zahlen y anhand des Graphen von f bestimmbar sind. Damit kann dann von Zeit zu getestet werden, ob die zu lernende Funktion f auf dem gerade vorliegenden Anfangsstück der Bedingung $\varphi_{f(1)}(x) = f(x)$ genügt. Solange dies der Fall ist, wird $(f(1), 1)$ auf Anfangsstücken ungerader Länge ausgegeben. Auf allen anderen Anfangstücken f^n geben

wir eine kanonische Nummer von $f^n 0^\infty$ aus, solange nicht bewiesen werden konnte, daß f *nicht* zu einer der möglichen Varianten der Klasse $U_{f(0)}$ gehört. Im letzten Fall produzieren wir nur noch Fehlermeldungen.

Wir kommen jetzt zur formalen Konstruktion. Dazu sei $i \in \mathbb{N}$. Die Klassen U_i werden uniform wie folgt definiert. Fixiere eine streng monoton wachsende Funktion $r \in \mathcal{R}$, so daß $\Phi_i = \varphi_{r(i)}$ für alle $i \in \mathbb{N}$ gilt (cf., Blum, 1967). Dann ist $Val\ r$ rekursiv. Möge $g \in \mathcal{R}$ so gewählt sein, daß für alle $j \in \mathbb{N}$ folgendes gilt:

$$\varphi_{g(j)}(0) = i$$

$$\varphi_{g(j)}(1) = \begin{cases} k, & \text{falls es } k \text{ mit } r(k) = j \text{ gibt,} \\ 0, & \text{sonst.} \end{cases}$$

$$\varphi_{g(j)}(2) = 0.$$

Die Definition von $\varphi_{g(j)}$ wird in Stufen fortgesetzt. Am Beginn von Stufe s möge $\varphi_{g(j)}(z)$ schon für alle $z \leq \sigma_{s-1}$ definiert sein, wobei $\sigma_0 = 2$. Für $s \geq 1$ definieren wir:

Stufe s. Sei $\tau = \varphi_{g(j)}^{\sigma_{s-1}}$. Für $\mu = 1, 2, 3, \ldots$. berechne parallel $m_\mu = M_i(\tau 0^\mu)$ und $\varphi_{m_\mu}(\sigma_{s-1} + \mu + 1)$ solange, bis ein $\varphi_{m_\mu}(\sigma_{s-1} + \mu + 1)$ sich als definiert erweist. Wird kein derartiges μ gefunden, so terminiert Stufe s nicht.

Angenommen, es wird ein μ mit $\varphi_{m_\mu}(\sigma_{s-1} + \mu + 1) \downarrow$ gefunden. Dann definieren wir

$$\varphi_{g(j)}(x) = \begin{cases} 0, & \text{falls } \sigma_{s-1} < x \leq \mu + \sigma_{s-1}, \\ \varphi_{m_\mu}(\sigma_{s-1} + \mu + 1) + 1, & \text{falls } x = \mu + \sigma_{s-1} + 1, \\ max\{\varphi_j(z) \mid z \leq \sigma_{s-1} + \mu + 1\} + 1, & \text{falls } x = \mu + \sigma_{s-1} + 2, \varphi_j(z) \downarrow, \\ & \text{für alle } z < x, \\ \text{nicht definiert,} & \text{falls } x = \mu + \sigma_{s-1} + 2, \varphi_j(z) \uparrow, \\ & \text{für ein } z < x, \\ 0, & \text{falls } x = \mu + \sigma_{s-1} + 3. \end{cases}$$

Beende dann Stufe s, setze $\sigma_s = \sigma_{s-1} + \mu + 3$ und beginne Stufe $s + 1$.

Nach dem Fixpunktsatz (cf., Rogers, 1967) kann effektiv eine Zahl b mit $\varphi_{g(r(b))} = \varphi_b$ gefunden werden. Unsere Konstruktion liefert dann sofort $\varphi_b(0) = i$. Diese Information wird später als Information über die zu simulierende Maschine genutzt. Weiterhin erhalten wir aus der Konstruktion unmittelbar $\varphi_b(1) = b$ und $\varphi_b(2) = 0$.

Hilfssatz. Wenn Stufe s verlassen wird, so ist $\varphi_b(z)$ für alle $z \leq \sigma_s$ definiert.

Wir beweisen den Hilfssatz induktiv. Damit Stufe s verlassen werden kann, muß ein μ derart gefunden worden sein, daß $\varphi_{m_\mu}(\sigma_{s-1}+\mu+1)$ sich als definiert herausgestellt hat. Damit ist nach Konstruktion $\varphi_b(x)=0$ für alle $x \in \{\sigma_{s-1}+1,\ ...,\ \sigma_{s-1}+\mu\}$. Nach Induktionsvoraussetzung ist somit $\Phi_b(x)$ für alle $x \leq \sigma_{s-1}+\mu$ definiert. Nun ist weiterhin $\varphi_b(\sigma_{s-1}+\mu+1) = \varphi_{m_\mu}(\sigma_{s-1}+\mu+1)+1$ und folglich definiert. Also konvergiert auch $\Phi_b(\sigma_{s-1}+\mu+1)$. Hieraus können wir schließen, daß $max\{\varphi_{r(b)}(z) \mid z \leq \sigma_{s-1}+\mu+1\}$ existiert. Damit ist auch $\varphi_b(\sigma_{s-1}+\mu+2)$ definiert. Offensichtlich ist nach Konstruktion $\varphi_b(\sigma_{s-1}+\mu+3)=0$ und deshalb konvergent. Da $\sigma_s = \sigma_{s-1}+\mu+3$ ist der Hilfssatz somit bewiesen.

q.e.d. (Hilfssatz)

Wird Stufe s für alle $s \in \mathbb{N}$ verlassen, so setzen wir $U_i = \{\varphi_b\}$. Nach dem gerade bewiesenen Hilfssatz ist dann $\varphi_b \in \mathcal{R}$. Andererseits produziert die IIM M_i auf φ_b nach Konstruktion unendlich viele Hypothesen m_j mit $\varphi_{m_j} \neq \varphi_b$. Also gilt $U_i \not\subseteq BC(M_i)$.

Sollte Stufe s nur für endlich viele $s \in \mathbb{N}$ terminieren, so haben wir folgende Situation. Die Stufen 1, 2, ..., s' terminieren, Stufe $s'+1$ jedoch nicht. Wir setzen $\tau = \varphi_b^{\sigma_{s'}}$ und $U_i = \{\tau 0^\infty\}$. Nach Konstruktion sind somit, bei sukzessiver Eingabe von $\tau 0^\infty$, fast alle Hypothesen von M_i nicht korrekt. Also gilt auch in diesem Fall $U_i \not\subseteq BC(M_i)$.

Insgesamt ergibt sich also $U \notin BC$ für $U = \bigcup_{i \in \mathbb{N}} U_i$. Es sei noch angemerkt, daß der Fixpunkt b effektiv aus der Kenntnis eines Programms für M_i berechenbar ist. Diesen Sachverhalt nutzen wir jetzt beim Entwurf einer IIM M aus, die U im Sinne von $\mathcal{TF}\text{-}REX_{freq}(\frac{1}{2})$ lernt.

Möge $t \in \mathcal{TF}$ und $n \in \mathbb{N}$ sein. Wir definieren:

$M(t^n) =$ „Wenn $n=0$ so gib $(f(0),1)$ aus.

Andernfalls verfahre wie folgt: Berechne den entsprechenden Fixpunkt b aus $M_{f(0)}$. Ist $n=1$, so teste, ob $f(1)=b$. Wenn ja, gib $(b,1)$ aus. Ist $f(1) \neq b$, so gib $(1,0)$ aus und setze FEHLER $=1$.

Wenn $n>1$, so teste, ob FEHLER $=1$. Wenn ja, so gib $(n,0)$ aus und gehe nach (A). Andernfalls gib $(f(1),1)$ bzw. $(e_n,1)$ aus, falls n gerade bzw. ungerade ist, wobei e_n ein kanonisches Programm der folgenden Funktion η_n ist. Sei z die größte Zahl $x \leq n$, für die $f(x) \neq 0$. Definiere

$$\eta(y) = \begin{cases} f(y), & \text{falls } y \leq z, \\ 0, & \text{falls } y > z. \end{cases}$$

Bevor der nächste Wert von f (d.h. $f(n+1)$) gelesen wird, arbeite die nachstehende Prozedur **ERROR** ab."

ERROR

(1) Teste, ob es eine Zahl y mit $0 \leq y \leq n-2$ derart gibt, daß alle Werte $f(y),\ f(y+1),\ f(y+2)$ ungleich Null sind. Wird ein deartiges y gefunden, so setze FEHLER $= 1$ und stoppe.

(2) Teste, ob es ein y mit $0 < y \leq n-1$ und $f(y) \neq 0$ aber $f(y-1) = f(y+1) = 0$ gibt. Wenn ja, so setze FEHLER $= 1$ und stoppe.

(3) Teste parallel für $x \in \{0, 1, ..., n\}$ mit $\Phi_{f(1)}(x) \leq n$, ob $\varphi_{f(1)}(x) = f(x)$. Wird ein x mit $\Phi_{f(1)}(x) \leq n$ und $\varphi_{f(1)}(x) \neq f(x)$ gefunden, so setze FEHLER $= 1$ und stoppe.

(4) Teste, ob es ein $y \leq n-2$ mit $f(y) \neq 0 \neq f(y+1)$ und $f(y-1) = 0 = f(y+2)$ gibt.
Für das größte derartige y teste weiterhin, ob $\Phi_{f(1)}(x) \leq f(x+1)$ für alle $x \leq y$.
Ist die Ungleichung für ein $x \leq y$ nicht erfüllt, so setze FEHLER $= 1$ und stoppe.
Andernfalls teste, ob $\varphi_{f(1)}(x) = f(x)$ für alle $x \leq y$.
Bemerkung: $\varphi_{f(1)}(x)$ ist für alle $x \leq y$ definiert, da $\Phi_{f(1)}(x) \leq f(x+1)$.
Wird ein $x \leq y$ mit $\varphi_{f(1)}(x) \neq f(x)$ gefunden, so setze FEHLER $= 1$ und stoppe.

end **ERROR**

(A) Lies den nächsten Wert von f.

Wir haben noch zu zeigen, daß $U \subseteq \mathcal{TF}\text{-}REX_{freq}(\frac{1}{2})(M)$. Sei $f \in \mathcal{TF}$. Ist $f \in U$, so kann nach Konstruktion nie FEHLER $= 1$ gesetzt werden. Folglich sind in diesem Fall fast alle Hypothesen von der Form $(j_n, 1)$.

Mögen jetzt fast alle Hypothesen eine 1 in der zweiten Komponente enthalten. Wir zeigen, daß dann eine für f korrekte Hypothese $(j, 1)$ existiert, die mit Häufigkeit $1/2$ in der Folge $(M(f^n))_{n\in\mathbb{N}}$ auftritt. Aus der Konstruktion ergibt sich unmittelbar, daß dann FEHLER $\neq 1$ gelten muß, da andernfalls falls alle Hypothesen von der Form $(n, 0)$ wären.

Fall 1. $\varphi_{f(1)} \in \mathcal{R}$.

Dann wird in Anweisung (4) der Prozedur **ERROR** $\varphi_{f(1)} = f$ verifiziert. Um dies zu sehen nehmen wir an, es gäbe ein x mit $\varphi_{f(1)}(x) \neq f(x)$. Dann wird jedoch in (3)

nach Lesen des Anfangsstückes der Länge $\Phi_{f(1)}(x)+1$ die Ungleichheit $\varphi_{f(1)}(x) \neq f(x)$ festgestellt und FEHLER = 1 gesetzt. Danach wären alle Hypothesen von der Form $(n,0)$, im Widerspruch zur Voraussetzung. Also wird für alle geraden n eine korrekte Hypothese ausgegeben.

Fall 2. $\varphi_{f(1)} \notin \mathcal{R}$.

Nach Konstruktion gibt es dann ein k_0, so daß $\varphi_b(x) = \varphi_{f(1)}(x)$ für alle $x \leq k_0$ und $\varphi_{f(1)}(x) \uparrow$ für alle $x > k_0$. Somit verifiziert M nach $max\{\Phi_{f(1)}(x) \mid x \leq k_0\}$ Schritten die Gültigkeit von $\varphi^{k_0}_{f(1)} = f^{k_0}$, da andernfalls wieder FEHLER = 1 gesetzt wird. Aus der Konstruktion sieht man jetzt unmittelbar, daß $f(x) = 0$ für alle $x > k_0$ gelten muß, da sonst FEHLER = 1 gesetzt wird. Also ist für alle ungeraden n die Hypothese $(e_n,0)$ korrekt. Folglich gilt $U \subseteq \mathcal{TF}\text{-}REX_{freq}(\frac{1}{2})(M)$.

q.e.d.

Der eben gezeigte Satz liefert nun unmittelbar unendliche Hierarchien für die zuverlässige Häufigkeitserkennung, die bei $\mathcal{M}\text{-}REX$ bzw. $\mathcal{M}\text{-}RBC$ beginnen, wobei $\mathcal{M} \in \{\mathcal{TF}, \mathcal{R}\}$ ist. Es bleibt also noch zu klären, wo diese Hierarchien enden. L. Harrington konnte die BC^*–Lernbarkeit der Klasse $\mathcal{R}$ nachweisen (cf., Case und Smith, 1983), woraus sich sofort $\mathcal{R} \in \mathcal{R}\text{-}RBC^*$ ergibt. Somit muß für alle $n \in \mathbb{N}$ sogar $\mathcal{R}\text{-}RBC_{freq}(\frac{1}{n}) \subset \mathcal{R}\text{-}RBC^*$ gelten, da nach Satz 3.17 $\mathcal{R} \notin \mathcal{R}\text{-}RBC_{freq}(\frac{1}{n})$ ist. Was gilt nun im Falle $\mathcal{M} = \mathcal{TF}$? Diese Frage beantworten wir nahezu vollständig mit den nächsten beiden Sätzen. Zur Diskussion der offengebliebenen Probleme verweisen wir auf Abschnitt 3.5.

Satz 3.18. $EX \# \mathcal{TF}\text{-}RBC^*$

Beweis: Nach Corollar 3.15 wissen wir schon, daß $\mathcal{TF}\text{-}RBC^* \setminus EX \neq \emptyset$. Den Nachweis $EX \setminus \mathcal{TF}\text{-}RBC^* \neq \emptyset$ führen wir in zwei Schritten. Zuerst zeigen wir $\mathcal{R} \notin \mathcal{TF}\text{-}RBC^*$. Dann wenden wir Satz 3.2 auf die Klasse $U = \{f \mid f \in \mathcal{R},\ \varphi_{f(0)} = f\}$ an, die offenbar im Limes gelernt werden kann.

Hilfssatz. $\mathcal{R} \notin \mathcal{TF}\text{-}RBC^*$.

Sei M irgendeine auf $\mathcal{TF}$ BC^*–zuverlässig arbeitende IIM. Möge $(\alpha_i)_{i\in\mathbb{N}}$ eine beliebig fixierte rekursive Aufzählung aller endlichen Funktionen sein, die auf Anfangsstücken $\{0,\ ...,\ n\}$ der Menge $\mathbb{N}$ definiert sind. Wir definieren eine rekursive Funktion f_M, so daß $\{f_M\} \notin \mathcal{TF}\text{-}RBC^*$ vermöge folgender Prozedur. Suche das kleinste k_1 mit $M(\alpha_{k_1}) = (j,0)$. Wenn dieses k_1 gefunden wurde, so setze $f_M(x) = \alpha_{k_1}(x)$ für alle $x \in Arg\ \alpha_{k_1}$. Sei $n_1 = max\{x \mid x \in Arg\ \alpha_{k_1}\}$. Wird kein k_1 gefunden, so ist f_M die nirgends definierte Funktion.

Angenommen, k_1 wurde gefunden. Wir iterieren die Konstruktion wie folgt. Suche das kleinste k_2 mit $M(\alpha_{k_1}\alpha_{k_2}) = (e, 0)$. Sollte ein derartiges k_2 gefunden werden, so definieren wir $f_M(n_1 + x) = \alpha_{k_2}(x)$ für alle $x \in Arg\ \alpha_{k_2}$ und setzen $n_2 = n_1 + max\{x | x \in Arg\ \alpha_{k_2}\}$. Wird kein k_2 gefunden, so wird f_M nicht weiter definiert. Iteriere diese Konstruktion.

Wir zeigen jetzt, daß $f_M \in \mathcal{R}$. Dazu möge $t \in \mathcal{TF} \setminus \mathcal{R}$ sein. Somit gilt $\varphi_i \neq t$ für alle $i \in \mathbb{N}$. Nach Voraussetzung arbeitet M BC^*–zuverlässig auf $\mathcal{TF}$. Folglich müssen in der Folge $(M(t^n))_{n\in\mathbb{N}}$ unendlich viele Fehlermeldungen auftreten. Sei $(M(t^{n_j}))_{n_j\in\mathbb{N}}$ die durch die Bedingung $M(t^{n_j}) = (e_{n_j}, 0)$ definierte Teilfolge. Wir interpretieren die endlichen Folgen $\beta_j = t(n_j + 1)...t(n_{j+1})$ in kanonischer Weise als endliche Funktionen, die auf $\{0, ..., n_{j+1} - n_j\}$ definiert sind. Da alle diese Funktionen β_j in der Aufzählung $(\alpha_i)_{i\in\mathbb{N}}$ vorkommen müssen, sehen wir, daß die Suche nach den entsprechenden k_j stets erfolgreich terminiert. Also ist $f_M \in \mathcal{R}$.

q.e.d. (Hilfssatz)

Sei jetzt $U = \{f | f \in \mathcal{R},\ \varphi_{f(0)} = f\}$. Angenommen, $U \in \mathcal{TF}{-}RBC^*$. Nach Satz 3.2 muß dann auch $[U] \in \mathcal{TF}{-}RBC^*$ gelten. Da $[U] = \mathcal{R}$, erhalten wir somit einen Widerspruch.

q.e.d.

Der letzte Satz dieses Kapitels liefert uns noch die Erkenntnis, daß zuverlässige Häufigkeitserkennung stets von geringerer Leistungsfähigkeit als zuverlässige BC^* Inferenz ist.

Satz 3.19. *$\mathcal{TF}{-}RBC_{freq}(\frac{1}{n}) \subset \mathcal{TF}{-}RBC^*$ für alle $n \geq 1$.*

Beweis: Nach Pitt (1989a) ist $BC_{team}(n) = BC_{freq}(\frac{1}{n})$ und $EX_{freq}(\frac{1}{n}) \subset BC_{freq}(\frac{1}{n})$ für alle $n \in \mathbb{N}$. Offenbar gilt weiterhin $\mathcal{TF}{-}REX_{freq}(\frac{1}{n}) \subset BC_{freq}(\frac{1}{n})$. Nach Satz 3.17 wissen wir auch, daß $\mathcal{TF}{-}REX_{freq}(\frac{1}{n+1}) \setminus BC_{team}(n) \neq \emptyset$ für alle $n \geq 1$. Folglich gilt $\mathcal{TF}{-}REX_{freq}(\frac{1}{n+1}) \setminus \mathcal{TF}{-}RBC_{freq}(\frac{1}{n}) \neq \emptyset$ und somit erst recht $\mathcal{TF}{-}RBC_{freq}(\frac{1}{n+1}) \setminus \mathcal{TF}{-}RBC_{freq}(\frac{1}{n}) \neq \emptyset$. Also ist $\mathcal{TF}{-}RBC_{freq}(\frac{1}{n}) \subset \mathcal{TF}{-}RBC_{freq}(\frac{1}{n+1})$. Deshalb ist zum Nachweis von $\mathcal{TF}{-}RBC_{freq}(\frac{1}{n}) \subset \mathcal{TF}{-}RBC^*$, nur noch $\mathcal{TF}{-}RBC_{freq}(\frac{1}{n}) \subseteq \mathcal{TF}{-}RBC^*$ für alle $n \geq 1$ zu zeigen.

Seien $n \in \mathbb{N}$ und $U \in \mathcal{TF}{-}RBC_{freq}(\frac{1}{n})$. Möge M eine beliebig fixierte IIM mit $U \subseteq \mathcal{TF}{-}RBC_{freq}(\frac{1}{n})(M)$ sein. Wir definieren die gewünschte IIM $\hat{M}$ mit $U \subseteq \mathcal{TF}{-}RBC^*(\hat{M})$ wie folgt. Bei Eingabe eines Anfangstückes f^n einer Funktion $f \in \mathcal{TF}$ berechnet $\hat{M}$ zunächst $M(f^n) = (j_n, b_n)$. Wenn $b_n = 0$, so gibt $\hat{M}$ das Paar $(n, 0)$ als Hypothese aus. Ist $b_n = 1$, so gibt $\hat{M}$ die Hypothese $(e_n, 1)$ aus, wobei e_n gerade das Programm ist, das L. Harringtons Maschine bei Eingabe von f^n berechnet (cf., Case und Smith, 1983). Es macht nun keine Mühe, $U \subseteq \mathcal{TF}{-}RBC^*(\hat{M})$ zu verifizieren.

q.e.d.

Im folgenden Bild haben wir die erhaltenen Kenntnisse bzgl. der Leistungsfähigkeit der zuverlässigen Häufigkeitserkennung zusammengefaßt.

Häufigkeitserkennung versus zuverlässige Häufigkeitserkennung

$$
\begin{array}{ccccccccc}
EX_{freq}(1) & \subset & EX_{freq}(\frac{1}{2}) & \subset \ldots \subset & EX_{freq}(\frac{1}{n}) & \subset & BC^* \\
\cup & & \cup & & \cup & & \cup| \\
\mathcal{M}-REX_{freq}(1) & \subset & \mathcal{M}-REX_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{M}-REX_{freq}(\frac{1}{n}) & \subset & \mathcal{M}-RBC^* \\
\cap & & \cap & & \cap & & \| \\
\mathcal{M}-RBC_{freq}(1) & \subset & \mathcal{M}-RBC_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{M}-RBC_{freq}(\frac{1}{n}) & \subset & \mathcal{M}-RBC^* \\
\cap & & \cap & & \cap & & \cap| \\
BC_{freq}(1) & \subset & BC_{freq}(\frac{1}{2}) & \subset \ldots \subset & BC_{freq}(\frac{1}{n}) & \subset & BC^*
\end{array}
$$

Abbildung 5

Wie bereits erwähnt, blieben einige Fragen offen. Wir gehen jetzt näher darauf ein.

3.5. Diskussion der Resultate und offene Probleme

Wir wollen jetzt die bislang vorgestellten Resultate hinsichtlich ihrer Bedeutung für die algorithmische Lerntheorie diskutieren. Ausgehend von den bereits von Blum und Blum (1975) sowie Minicozzi (1976) erzielten Resultaten haben wir zunächst die Leistungsfähigkeit deterministischer zuverlässiger Lernalgorithmen in Abhängigkeit von der Anzahl der erlaubten Anomalien und der zugrunde liegenden Zuverlässigkeitsmenge $\mathcal{M}$ detailliert untersucht. Mit den Sätzen 3.4 und 3.6 ist es erstmals gelungen, den Einfluß erlaubter Anomalien genau zu erforschen. Dabei zeigte es sich, daß bereits eine fixierte endliche Anzahl a von zugelassenen Fehlern die Lernfähigkeit um bis dahin unbekannte Größenordnungen erweitern kann. Bislang war nämlich lediglich bekannt, daß $EX^a \subset EX^{a+1} \subset EX_{team}(a+2)$, $a \in \mathbb{N}$, und $\mathcal{R}-REX \subset \mathcal{R}-REX^*$ gilt. Zieht man nun zusätzlich in die Beziehung $\mathcal{TF}-REX^a \subset EX_{team}(a+1)$ in Betracht so sieht man aus Satz 3.4 sofort, daß der Zuwachs an Leistungsfähigkeit die theoretisch größtmögliche Grenze erreichen kann. Wie Satz 3.6 zeigt, können aber erlaubte Anomalien nicht als universelles Mittel angesehen werden, um möglichst leistungsfähige Lernalgorithmen zu entwerfen. Weiterführende Untersuchungen sollten deshalb vor allem der Charakterisierung von Funktionenklassen dienen, die erst durch erlaubte Anomalien in einem bestimmten Sinne erlernbar werden. Satz 3.5 stellt ein erstes Resultat in diese Richtung dar.

Es ist weiterhin gelungen, den Begriff des zuverlässigen Lernens in Bereiche auszudehnen, die sich bislang allen derartigen Versuchen erfolgreich widersetzt hatten. Wir konnten somit Möglichkeiten aufzeigen, wie Pluralismus in adäquater Form in den Bereich des zuverlässigen Lernens integrierbar ist. Das gelang einerseits dadurch, daß in den Definitionen 11 und 8 ein begrenzter Nichtdeterminismus zugelassen wurde. Mit der Definition 11 stellten wir neue probabilistische Lerntypen vor, die erstmals den Begriff des Lernens mit einseitiger Fehlerwahrscheinlichkeit in nicht trivialer Weise in die induktive Inferenz übertrugen. Die in Kapitel 3 vorgestellten Resultate liefern aus unserer Sicht hinreichend viel Evidenz dafür, daß mit den eingeführten Begriffsbildungen das Wesen zuverlässiger Lernalgorithmen richtig erfaßt wurde. Wie die Sätze 3.1 und 3.2 zeigen, konnten nämlich gerade die Eigenschaften gerettet werden, die deterministisches zuverlässiges Lernen attraktiv gemacht hatten. Darüber hinaus erweitern die vorgestellten Konzepte pluralistischer zuverlässiger Inferenz die Lernfähigkeit zum Teil beträchtlich. Insgesamt konnten vier neue unendliche Hierarchien aufgedeckt werden. Mit Corollar 3.10 und 3.13 gelang weiterhin der Nachweis, daß es egal ist, ob Pluralismus in Form eines begrenzten Nichtdeterminismus oder durch probabilistische Hilfsmittel im Lernprozeß realisiert wird.

Alle neuen Lerntypen wurden außerdem zu früher studierten Inferenzmodi in Beziehung gesetzt. Damit konnten die prinzipiellen Stärken und Schwächen der neuen Lerntypen transparenter gemacht werden. Es blieben jedoch einige wichtige Fragen offen. Wir vermuten, daß sich die topolgischen Eigenschaften der Zuverlässigkeitsmenge $\mathcal{TF}$ auch im Falle der zuverlässigen Häufigkeitserkennung auswirken werden. Es gelang jedoch nicht, die Inklusionen $\mathcal{TF}-REX_{freq}(1/n) \subset \mathcal{R}-REX_{freq}(1/n)$ bzw. $\mathcal{TF}-RBC_{freq}(1/n) \subset \mathcal{R}-RBC_{freq}(1/n)$ zu beweisen, sofern $n \geq 2$ ist. Es möglicher Weg zur Verifikation bzw. Falsifizierung der genannten Inklusionen wäre, die genannten Lerntypen zunächst in komplexitätstheoretischen Termini zu charakterisieren und dann ähnlich wie im Beweis von Satz 3.6 zu verfahren. Derartige Charakterisierungen oder Charakterisierungen in numerierungstheoretischen Termini wären darüber hinaus aber auch von generellem Interesse. Wie die Arbeiten von Blum und Blum (1975), Wiehagen (1977, 1991), Jain und Sharma (1989) und Zeugmann (1983) zeigen, erlauben Charakterisierungssätze häufig vertiefte und unifizierende Einblicke in die Arbeitsweise von Lernalgorithmen.

Ein weiteres interessantes offenes Problem betrifft die verhaltenskorrekte zuverlässige Inferenz mit erlaubten Anomalien auf den Mengen $\mathcal{R}$ und $\mathcal{TF}$. Vermöge einer geeigneten Modifikation der von Daley (1983) entwickelten Beweistechniken kann man zwar $\mathcal{M}-RBC^a \subseteq \mathcal{M}-RBC_{freq}(1/(a+1))$ für alle $a \in \mathbb{N}$ beweisen. Wir vermuten jedoch, daß

sogar die echte Inklusion gilt. Weiterhin blieb offen, ob $\mathcal{M}-RBC^a \subset \mathcal{M}-RBC^{a+1}$ für $\mathcal{M} \in \{\mathcal{TF}, \mathcal{R}\}$ und $a \in \mathbb{N}$ gilt. Es liegt auf der Hand, daß ein BC-Analogon zu Theorem 3.4 die Frage bejahend beantwortete. Der Autor muß jedoch eingestehen, daß ihm die Beantwortung der Frage, ob $\mathcal{TF}-RBC^a \setminus BC_{team}(a) \neq \emptyset$ noch nicht gelungen ist. Schließlich seien noch zwei interessante Fragen genannt, die sich direkt aus den Arbeiten von Chen (1981, 1982) ergeben. Das erste Problem betrifft die Güte der im Limes erlernten Programme bzgl. ihrer Größe. Da Programme ja gespeichert werden müssen, ist man daran interessiert, möglichst Programme zu lernen, deren Größe von der minimal erreichbaren höchstens um einen rekursiven Faktor abweicht. Dieses Lernziel ist nun im allgemeinen nicht zu erreichen, wenn keine Anomalien erlaubt werden (cf., Freivalds, 1990). Wie Chen (1982) nachwies, ändert sich die Situation grundlegend, wenn man bereit ist, endlich viele Anomalien in Kauf zu nehmen. Dann lassen sich für alle im Sinne von EX erlernbaren Funktionenklassen U stets IIMs angeben, die U im Sinne von EX^* derart lernen, daß die im Limes inferierten Programme minimale Länge (modulo eines rekursiven Faktors) haben. Es entsteht natürlich die Frage, ob sich dieses Resultat inhaltlich auf die zuverlässige Häufigkeitserkennung im Sinne von EX ausdehnen läßt. Das zweite Problem ist eng mit der Verbandsstruktur der Familien EX^a bzw. $\mathcal{M}-REX^a$, $\mathcal{M}-RBC^a$, $\mathcal{M}-REX_{freq}(1/a)$ und $\mathcal{M}-RBC_{freq}(1/a)$ verbunden. Chen (1981) konnte nähmlich zeigen, daß sich jede EX^a–erlernbare Klasse U in eine Klasse $\hat{U}$ mit $\hat{U} \in EX^{a+1} \setminus EX^a$ einbetten läßt. Somit ergibt sich die Frage, ob dieses Resultat auch für alle neu eingeführten Typen des zuverlässigen Lernens gilt.

Abschließend weisen wir noch darauf hin, daß es sich natürlich anbietet, zuverlässiges Lernen von rekursiv aufzählbaren Sprachen zu untersuchen. Wie Osherson et al. (1986) zeigen konnten, ist hier allerdings nur der Fall des Lernens bei Informantbelehrung von Interesse. Dabei sollte man als Lernziel die Synthese von Generatoren ins Auge fassen. Als Zuverlässigkeitsmengen bieten sich dann kanonisch die Menge aller Sprachen, aller rekursiv aufzählbaren Sprachen sowie die Menge aller rekursiven Sprachen an.

Damit wollen wir den Ausblick auf weitere Forschungen abschließen. Im letzten Abschnitt dieses Kapitels stellen wir die erhaltenen Resultate noch einmal in Bildform zusammenfassend vor.

3.6. Zusammenfassung

In diesem Abschnitt stellen wir die vielfältigen Inklusionsbeziehungen zwischen Monte-Carlo Inferenz, zuverlässiger Häufigkeitserkennung und algorithmischem Lernen durch

Teams in übersichtlicher Form zusammen. Mit dem folgenden Bild charakterisieren wir die Beziehungen zwischen der Leistungsfähigkeit von Monte-Carlo Inferenzmaschinen und zuverlässiger EX– bzw. BC–Inferenz vollständig.

Monte-Carlo Inferenz und zuverlässige Häufigkeitserkennung

$$
\begin{array}{ccccccccc}
\mathcal{TF}\text{-}REX_{freq}(1) & \subset & \mathcal{TF}\text{-}REX_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{TF}\text{-}REX_{freq}(\frac{1}{n}) & \subset \ldots \\
\| & & \| & & \| & \\
\mathcal{TF}\text{-}REX_{prob}(1) & \subset & \mathcal{TF}\text{-}REX_{prob}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{TF}\text{-}REX_{prob}(\frac{1}{n}) & \subset \ldots \\
\cap & & \cap| & & \cap| & \\
\mathcal{R}\text{-}REX_{prob}(1) & \subset & \mathcal{R}\text{-}REX_{prob}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{R}\text{-}REX_{prob}(\frac{1}{n}) & \subset \ldots \\
\| & & \| & & \| & \\
\mathcal{R}\text{-}REX_{freq}(1) & \subset & \mathcal{R}\text{-}REX_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{R}\text{-}REX_{freq}(\frac{1}{n}) & \subset \ldots \\
\cap & & \cap & & \cap & \\
\mathcal{R}\text{-}RBC_{freq}(1) & \subset & \mathcal{R}\text{-}RBC_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{R}\text{-}RBC_{freq}(\frac{1}{n}) & \subset \ldots \\
\| & & \| & & \| & \\
\mathcal{R}\text{-}RBC_{prob}(1) & \subset & \mathcal{R}\text{-}RBC_{prob}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{R}\text{-}RBC_{prob}(\frac{1}{n}) & \subset \ldots \\
\cup & & \cup| & & \cup| & \\
\mathcal{TF}\text{-}RBC_{prob}(1) & \subset & \mathcal{TF}\text{-}RBC_{prob}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{TF}\text{-}RBC_{prob}(\frac{1}{n}) & \subset \ldots \\
\| & & \| & & \| & \\
\mathcal{TF}\text{-}RBC_{freq}(1) & \subset & \mathcal{TF}\text{-}RBC_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{TF}\text{-}RBC_{freq}(\frac{1}{n}) & \subset \ldots
\end{array}
$$

Abbildung 6

Im folgenden Bild vergleichen wir die Leistungskraft von zuverlässigen Lernalgorithmen und Teaminferenz.

Zuverlässiges Lernen und seine Beziehungen zur Teaminferenz

$$
\begin{array}{ccccccccc}
\mathcal{M}-REX & \subset & \mathcal{M}-REX^1 & \subset \ldots \subset & \mathcal{M}-REX^a & \subset \ldots \subset & \mathcal{M}-REX^* \\
\cap & & \cap & & \cap & & \\
EX_{team}(1) & \subset & EX_{team}(2) & \subset \ldots \subset & EX_{team}(a+1) & \subset \ldots & \\
\| & & \| & & \| & & \\
EX_{freq}(1) & \subset & EX_{freq}(\frac{1}{2}) & \subset \ldots \subset & EX_{freq}(\frac{1}{a+1}) & \subset \ldots & \\
\# & & \# & & \# & & \\
\mathcal{M}-REX^1 & \subset & \mathcal{M}-REX^2 & \subset \ldots \subset & \mathcal{M}-REX^{a+1} & \subset \ldots \subset & \mathcal{M}-REX^* \\
\cap & & \cap & & \cap & & \cap \\
\mathcal{M}-RBC & \subset & \mathcal{M}-RBC^1 & \subset \ldots \subset & \mathcal{M}-RBC^a & \subset \ldots \subset & \mathcal{M}-RBC^* \\
\cap| & & \cap| & & \cap| & & \| \\
\mathcal{M}-RBC_{freq}(1) & \subset & \mathcal{M}-RBC_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{M}-RBC_{freq}(\frac{1}{a+1}) & \subset \ldots \subset & \mathcal{M}-RBC^* \\
\cup & & \cup & & \cup & & \| \\
\mathcal{M}-REX_{freq}(1) & \subset & \mathcal{M}-REX_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{M}-REX_{freq}(\frac{1}{a+1}) & \subset \ldots \subset & \mathcal{M}-RBC^* \\
\cap & & \# & & \# & & \\
BC & = & BC_{team}(1) & \subset \ldots \subset & BC_{team}(a) & \subset \ldots \subset & BC^* \\
\cap & & \# & & \# & & \\
\mathcal{M}-RBC_{freq}(1) & \subset & \mathcal{M}-RBC_{freq}(\frac{1}{2}) & \subset \ldots \subset & \mathcal{M}-RBC_{freq}(\frac{1}{a+1}) & \subset \ldots \subset & \mathcal{M}-RBC^*
\end{array}
$$

Abbildung 7

4. Algorithmisches Lernen von Sprachen

In diesem Kapitel studieren wir die im Abschnitt 2.3 eingeführten Lerntypen. Um den Umfang der vorliegenden Arbeit nicht zu sprengen, waren wir gezwungen, uns auf eine repräsentative Auswahl der erzielten Resultate zu beschränken. So werden wir auch nicht alle Beweise ausführen, sondern gegebenenfalls auf die Literatur verweisen. Unser Hauptaugenmerk gilt dem Lernen bei Textbelehrung. Wenn es um die Herausarbeitung der Stärken bzw. Schwächen des Lernens aus positiven Daten geht, werden wir auch Lernen bei Informantbelehrung betrachten. In der zweiten Sektion dieses Kapitels fassen wir die bislang erzielten Resultate bzgl. des Lernens aus positiven und negativen Daten zusammen. Die dritte Sektion ist Charakterisierungen gewidmet. Auch hier konzentrieren wir uns auf das Lernen bei Textbelehrung. In den ersten Abschnitten dieser Sektion geben wir Charakterisierungen für fast alle Typen des monotonen und dual monotonen Lernens an. Danach beschäftigen wir uns mit der Frage, warum beipielsweise exaktes Lernen schwächer sein kann als klassentreue Inferenz. Wir geben hinreichende und notwendige Bedingungen an, unter denen exaktes Lernen von gleicher Leistungsfähigkeit wie klassentreues Lernen ist. Schließlich fragen wir noch, welchen Typ von Hypothesen man beim klassenumfassenden Lernen in den Hypothesenraum aufnehmen sollte, um Lernalgorithmen höchster Leistungsfähigkeit zu erhalten. In der letzten Sektion dieses Kapitels diskutieren wir die erhaltenen Resultate, stellen offene Probleme vor und geben einen Ausblick auf weitere Forschungen.

4.1. Lernen bei Textbelehrung

Wir beginnen mit Lernen im Limes und fragen zunächst nach dem Einfluß der Wahl des Hypothesenraumes auf die prinzipielle Leistungsfähigkeit von im Limes lernenden induktiven Inferenzmaschinen. Danach studieren wir die gleiche Fragestellung unter Effizienzgesichtspunkten. Als Maß für die Effizienz wählen wir die Anzahl der Hypothesenwechsel. Dieses Effizienzmaß wurde von Gold (1967) eingeführt und seitdem intensiv studiert. Im dritten Abschnitt untersuchen wir den Einfluß der in Abschnitt 2.3 eingeführten Monotonieforderungen ausführlich.

4.1.1. Lernen im Limes und finite Inferenz

Der folgendene Satz besagt insbesondere, daß $ELIM\text{–}TXT = LIM\text{–}TXT = KLIM\text{–}TXT$ und $ELIM-INF = LIM-INF = KLIM-INF$ gilt.

Satz 4.1. *Seien $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ ein Hypothesenraum mit uniform entscheidbarem Wortproblem und $\mathcal{L} \subseteq \{L(G_j) | j \in \mathbb{N}\}$. Dann gilt: Wenn es eine IIM M gibt, die $\mathcal{L}$ bzgl. $\mathcal{G}$ bei Text- bzw. Informantbelehrung im Limes inferiert, so kann aus M eine IIM $\hat{M}$ konstruiert werden, die $\mathcal{L}$ exakt bei Text- bzw. Informantbelehrung im Limes lernt.*

Beweis: Die Beweise für Text– und Informantbelehrung sind nahezu identisch. Wir geben deshalb nur den Beweis für den Textfall an. Es sei noch darauf hingewiesen, daß der folgende Beweis teilweise auf einer schon in Gold (1967) (cf., Theorem 10.1., pp. 464) entwickelten Idee beruht. Sei $\mathcal{L} \subseteq LIM - TXT(M)$ bzgl. des Hypothesenraumes $\mathcal{G}$. Die gewünschte IIM $\hat{M}$ definieren wir wie folgt. Seien $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$. Wir setzen:

$\hat{M}(t_x) =$ „Simuliere $M(t_x)$. Gibt M auf t_x keine Hypothese aus, so gib keine Hypothese aus und lies die nächste Eingabe.
Andernfalls sei $j_x = M(t_x)$. Generiere das Anfangsstück i_x des lexikographisch geordneten Informanten i für $L(G_{j_x})$. Suche den kleinsten Index $k \leq x$, für den $i_x = i_x^{L_k}$ gilt, wobei $i_x^{L_k}$ das Anfangsstück der Länge $x+1$ vom lexikographisch geordneten Informanten i^{L_k} der Sprache $L_k \in \mathcal{L}$ bezeichnet. Wird dieses k gefunden, so gib k aus. Andernfalls gib keine Hypothese aus und lies die nächste Eingabe.“

Wir haben zu zeigen, daß $\mathcal{L} \in LIM - TXT(\hat{M})$ bzgl. $\mathcal{L}$. Da das Wortproblem für alle $L(G_j)$ uniform entscheidbar und $\mathcal{L}$ eine indizierte Familie ist, ergibt sich sofort die Rekursivität von $\hat{M}$. Also ist $\hat{M}$ eine IIM. Mögen jetzt $L \in \mathcal{L}$ und $t \in \mathit{Text}(L)$ sein. Nach Voraussetzung konvergiert die IIM M auf t gegen eine für L korrekte Hypothese j im Hypothesenraum $\mathcal{G}$, d.h. es gilt $M(t_x) = j$ und $L = L(G_j)$ für fast alle $x \in \mathbb{N}$. Somit existiert ein x_0 derart, daß $M(t_x) = j$ für alle $x \geq x_0$. Da $L \in \mathcal{L}$ ist, muß außerdem ein Index k_0 mit $k_0 = \mu z[L = L_z]$ existieren. Folglich gibt die Maschine $\hat{M}$ für alle $x \geq max\{x_0, k_0\}$ eine Hypothese aus. Schließlich folgt aus $k_0 = \mu z[L = L_z]$ unmittelbar, daß die IIM $\hat{M}$ jede Hypothese $k < k_0$ irgendwann verwerfen muß und anschließend $i_x = i_x^{L_{k_0}}$ verifiziert. Also konvergiert $\hat{M}$ auf t gegen eine für L korrekte Hypothese bzgl. des Hypothesenraumes $\mathcal{L}$.

q.e.d.

Der eben gegebene Beweis erhält allerdings nicht die Anzahl der Hypothesenwechsel. Im folgenden Abschnitt werden wir untersuchen, ob der Beweis von Satz 4.1 dahingehend verbessert werden kann, daß beim Übergang von einem Hypothesenraum zum anderen die Effizienz des Lernprozesses nicht negativ beinflußt wird. Wir setzen diesen Abschnitt mit

dem folgenden Satz fort, der insbesondere die Gleichheit zwischen exaktem, klassentreuem und klassenumfassendem finitem Lernen ergibt.

Satz 4.2. *Seien $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum mit uniform entscheidbarem Wortproblem und $\mathcal{L} \subseteq \{L(G_j) | j \in \mathbb{N}\}$. Dann gilt: Wenn es eine IIM M gibt, die $\mathcal{L}$ bzgl. $\mathcal{G}$ bei Text- bzw. Informantbelehrung finit inferiert, so kann aus M eine IIM $\hat{M}$ konstruiert werden, die $\mathcal{L}$ finit bei Text- bzw. Informantbelehrung bzgl. $\mathcal{L}$ lernt.*

Beweis: Wir führen den Beweis in zwei Schritten durch. Als erstes zeigen wir, wie mit Hilfe einer klassenumfassend lernenden IIM der im Lernprozeß genutzte Hypothesenraum in einen klassentreuen transformiert wird. Die konstruktive Gleichheit von klassentreuem und exaktem finiten Lernen wird im zweiten Schritt bewiesen, für den wir auf die Corollare 4.40 und 4.42 verweisen.

Da die Beweise für Text- und Informantbelehrung nahezu identisch sind, beschränken wir uns auf den Textfall. Mögen $\mathcal{L} = (L_j)_{j\in\mathbb{N}}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum und M eine IIM derart sein, daß $\mathcal{L}$ von M bzgl. $\mathcal{G}$ gelernt wird. Wir konstruieren jetzt wie folgt einen klassentreuen Hypothesenraum $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ für $\mathcal{L}$.

Sei $j \in \mathbb{N}$. Wir definieren $\tilde{G}_j$ folgendermaßen. Generiere den kanonischen Text t^j von L_j sukzessive bis das erste y gefunden wird, so daß M auf t^j_y eine Hypothese, sagen wir j_y, produziert. Dann setzen wir $\tilde{G}_j = G_{j_y}$ und $n(j) = y$.

Da M die Familie $\mathcal{L}$ bei Textbelehrung finit bzgl. $\mathcal{G}$ erlernt, muß für jedes $j \in \mathbb{N}$ ein $y \in \mathbb{N}$ gefunden werden, so daß M auf t^j_y eine Hypothese j_y produziert. Insbesondere ist j_y auch korrekt für L_j, woraus sich unmittelbar ergibt, daß $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ klassentreuer Hypothesenraum ist. Wir weisen weiter darauf hin, daß sich aus dem Gesagten auch sofort die Berechenbarkeit der Zahlen $n(j)$ ergibt.

Wir definieren nun eine IIM $\tilde{M}$, die $\mathcal{L}$ bzgl. $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ lernt. Seien $L \in \mathcal{L}$, $\tilde{t} \in \mathit{Text}(L)$ und $x \in \mathbb{N}$. Wir setzen:

$\tilde{M}(\tilde{t}_x) =$ „Berechne sukzessive für $j = 0, 1, ..., x$ die zugehörige Zahl $n(j)$ und teste, ob $Val\ t^j_{n(j)} \subseteq \tilde{t}^+_x$, wobei t^j der kanonische Text von $L(\tilde{G}_j)$ ist. Sobald der Test für ein j erfüllt ist, gib j aus und stoppe.
Ist der Test für kein $j \leq x$ erfüllt, so gib keine Hypothese aus und lies die nächste Eingabe."

Wir haben noch zu zeigen, daß $\tilde{M}$ das Verlangte leistet. Da $\tilde{\mathcal{G}}$ klassentreuer Hypothesenraum ist, wird von $\tilde{M}$ stets eine Hypothese ausgegeben. Aus $Val\ t^j_{n(j)} \subseteq \tilde{t}^+_x \subseteq L$ ergibt

sich weiterhin, daß $t^j_{n(j)}$ auch Anfangsstück eines Textes für die zu lernende Sprache L ist. Nach Konstruktion war aber $j_y = M(t^j_{n(j)}) = M(t^j_y)$ und $\tilde{G}_j = G_{j_y}$. Da M die Familie $\mathcal{G}$ finit erlernt, muß $L = L(G_{j_y}) = L(\tilde{G}_j)$ gelten, also ist $\mathcal{L} \subseteq FIN{-}TXT(\tilde{M})$ bzgl. $\tilde{\mathcal{G}}$.

q.e.d.

Wir beenden diesen Abschnitt mit folgendem Satz, der zeigt, daß aus positiven und negativen Daten mehr lernbar ist als alleine aus positiven Beispielen.

Satz 4.3.

(1) $LIM{-}TXT \subset LIM{-}INF$

(2) $FIN{-}TXT \subset FIN{-}INF$

Beweis: Wir beginnen mit Aussage (1). Dazu seien $L_0 = \{a\}^*$ und $L_k = L_0 \setminus \{a^k\}$ für $k \geq 1$. Möge $\mathcal{L} = L_0,\ L_1,\ L_2,\ \ldots$ die kanonische Aufzählung aller Sprachen L_k sein. Dann ist $\mathcal{L}$ offensichtlich eine indizierte Familie.

***Hilfssatz* A**. $\mathcal{L} \in LIM{-}INF$.

Die gewünschte IIM M definieren wir informal wie folgt. Initial gibt M stets 0 als Hypothese aus. Diese Hypothese wird solange wiederholt, bis ein $(a^k, -)$ im Informant gelesen wird. Die Maschine M wechselt dann die Hypothese von 0 zu k. Die Hypothese k wird dann auch in allen nachfolgenden Lernschritten ausgegeben. Offenbar lernt M die Familie $\mathcal{L}$ bei Informantbelehrung bzgl. $\mathcal{L}$.

q.e.d. (Hilfssatz A)

***Hilfssatz* B**. $\mathcal{L} \notin LIM{-}TXT$.

Zum Beweis dieses Hilfssatzes nutzen wir Angluins (1980a) Charakterisierungssatz für $LIM{-}TXT$ (cf., Satz 4.33). Dieser Satz besagt insbesondere, daß eine indizierte Familie $\mathcal{L}$ nur dann im Limes gelernt werden kann, wenn zu jeder Sprache $L \in \mathcal{L}$ eine rekursiv aufzählbare Tell-tale Menge T existiert, die folgende Bedingungen erfüllt:

(1) $T \subseteq L$ und

(2) für alle $j \geq 1$: $T \subseteq L_j$ impliziert $L_j \not\subset L$.

Somit genügt es zum Nachweis von $\mathcal{L} \notin LIM{-}TXT$ zu zeigen, daß L_0 keine endliche Tell-tale Menge besitzt. Angenommen doch, d.h. es gibt eine endliche rekursiv-aufzählbare Menge T, die (1) und (2) erfüllt.

Sei $z = max\{|w| \mid w \in T\}$. Aus der Definition der Sprachen L_k ergibt sich unmittelbar, daß $L_{z+1} \supset T$. Da außerdem nach Konstruktion $L_{z+1} \subseteq L_0$ ist, erhalten wir einen Widerspruch zu Bedingung (2).

q.e.d. (Hilfssatz B)

Die Aussage (2) des Satzes zeigt man mit Hilfe der folgenden indizierten Familie $\mathcal{L}_{dmon}$. Sei $L_k = \{a\}^* \setminus \{a^{k+1}\}$ für alle $k \in \mathbb{N}$ und $\mathcal{L}_{dmon} = (L_k)_{k \in \mathbb{N}}$. Diese Familie ist offensichtlich bei Informantbelehrung finit erlernbar. Wir beweisen jetzt nicht nur $\mathcal{L}_{dmon} \notin FIN\text{–}TXT$, sondern das wesentlich strengere Resultat $\mathcal{L}_{dmon} \notin KMON-TXT$.

Angenommen, $\mathcal{L}_{dmon} \in KMON-TXT$. Dann existieren eine IIM M und ein klassentreuer Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ mit $\mathcal{L}_{dmon} \subseteq KMON-TXT(M)$ bzgl. $\mathcal{G}$. Wir betrachten das Verhalten von M auf folgendem Text t_{fool}. M erhält $a^2, a^3, ...$ solange als Eingabe, bis die erste Hypothese j_{z-2}, sagen wir auf $a^2, a^3, ..., a^z$, mit $L(G_{j_{z-2}}) = L_0$ produziert wird. Ein derartiges z muß es geben, da andernfalls die Sprache L_0 nicht gelernt wird. Dann erhält M die Wörter $a, a^{z+2}, a^{z+3}, ...$ als Eingabe, bis eine korrekte Hypothese j_{r-2} für L_z ausgegeben wird. Da M die Sprache L_z bei Textbelehrung lernt, existiert folglich ein $r \geq z+2$, so daß M bei Eingabe von $a^2, a^3, ..., a^z, a, a^{z+2}, a^{z+3}, ..., a^r$ die Hypothese j_{r-2} ausgibt. Jetzt können wir dieses textuelle Anfangsstück aber mit $a^{z+1}, a^{r+2}, a^{r+3}, ...$ solange fortsetzen, bis eine korrekte Hypothese für L_r produziert wird. Der gewünschte Text t_{fool} wird dann wie folgt definiert:
$t_{fool} = a^2, a^3, ..., a^z, a, a^{z+2}, ..., a^r, a^{z+1}, a^{r+2}, a^{r+3}, ...$

Somit ist $t_{fool} \in \mathit{Text}(L_r)$. Da $a^{z+1} \in L_0 \cap L_r$ aber $a^{z+1} \notin L_z \cap L_r$ erhalten wir nun $L_0 \cap L_r \not\subseteq L_z \cap L_r$, womit M auf t_{fool} nicht monoton arbeitet.

q.e.d.

4.1.2. Lernen unter einer begrenzten Anzahl von Hypothesenwechseln

Wir studieren in diesem Abschnitt Lernen im Limes unter Effizienzforderungen. Als Maß für die Effizienz eines Lernalgorithmus nutzen wir die Anzahl der Hypothesenwechsel. Dieses Effizienzmaß ist insbesondere im Bereich der induktiven Inferenz rekursiver Funktionen ausführlich studiert worden. Barzdin und Freivalds (1972, 1974) bewiesen das folgende bemerkenswerte Resultat hinsichtlich der effizienten Lernbarkeit rekursiv aufzählbarer Klassen rekursiver Funktionen. Golds (1967) „Identifizierung durch Aufzählung“ erlaubt das Lernen der n-ten Funktion innerhalb der Aufzählung, erfordert jedoch $n-1$ Hypothesenwechsel. Es gibt aber einen Lernalgorithmus und einen Hypothesenraum, so daß sich die n-te Funktion der Aufzählung mit $\log n + o(\log n)$ vielen Hypothesenwechseln lernen läßt. Diese Schranke ist optimal. Insbesondere zeigt dieses Resultat, daß die Wahl des Hypothesenraumes die Effizienz des Lernprozesses dramatisch beeinflussen kann. Case und Smith (1983) etablierten eine Hierarchie von Lernverfahren in Abhängigkeit von der

Anzahl der höchstens erlaubten Hypothesenwechsel. Wiehagen, Freivalds und Kinber (1984) nutzten die Anzahl der Hypothesenwechsel um die Überlegenheit probabilistischer Lernfahren über deterministische zu beweisen. Somit erscheint es nur natürlich, danach zu fragen, ob dieses Effizienzmaß im Bereich des Lernens indizierter Familien von ähnlicher Bedeutung ist. Die Beantwortung dieser Frage ist nicht trivial, da sich induktive Inferenz rekursiver Funktionen deutlich vom Lernen rekursiv aufzählbarer Sprachen unterscheiden kann (cf., z.B. Case, 1988, Fulk, 1990). Wir knüpfen mit unseren Untersuchungen an die Arbeit von Mukouchi (1992b) an, in der exaktes Lernen unter einer vorgegebenen Anzahl von Hypothesenwechseln studiert wurde. Die Resultate dieses Abschnittes wurden vom Autor gemeinsam mit S. Lange erzielt (cf., Lange und Zeugmann, 1993a).

Satz 4.4. (Mukouchi, 1992b)

$$ELIM_0\text{–}TXT \subset ELIM_1\text{–}TXT \subset ELIM_2\text{–}TXT \subset ... \subset ELIM\text{–}TXT$$

Wir werden diesen Satz in mehrere Richtungen verschärfen. Zunächst haben wir:

Satz 4.5. $\bigcup_{k\in\mathbb{N}} LIM_k\text{–}TXT \subset LIM\text{–}TXT$

Beweis: Aus Definition 14 erhalten wir unmittelbar $\bigcup_{k\in\mathbb{N}} LIM_k\text{–}TXT \subseteq LIM\text{–}TXT$. Zum Nachweis von $LIM\text{–}TXT \setminus \bigcup_{k\in\mathbb{N}} LIM_k\text{–}TXT \neq \emptyset$ verwenden wir wieder die Familie $\mathcal{L}_{dmon}$ aus dem Beweis von Satz 4.3. $\mathcal{L}_{dmon} \in LIM\text{–}TXT$ ergibt sich unmittelbar aus dem nachstehenden Corollar, wo sogar schärfer $\mathcal{L}_{dmon} \in ECONSERVATIVE\text{–}TXT$ bewiesen wird.

Wir haben also noch zu zeigen, daß $\mathcal{L}_{dmon}$ nicht mit einer beliebig fixierten vorgegebenen Anzahl von Hypothesenwechseln erlernbar ist. Dazu sind keine anderen als die im Beweis von Satz 4.3, Aussage (2) verwendeten Ideen nötig. Man hat lediglich k Hypothesenwechsel zu erzwingen.

q.e.d.

Der Beweis des letzten Satzes erlaubt folgendes Corollar.

Corollar 4. 6. $ECONSERVATIVE\text{–}TXT \setminus \bigcup_{k\in\mathbb{N}} LIM_k\text{–}TXT \neq \emptyset$

Beweis: Die IIM M zum exakten konservativen Lernen von $\mathcal{L}_{dmon}$ definieren wir informal wie folgt. Seien $L \in \mathcal{L}_{dmon}$, $t \in Text(L)$ und $x \in \mathbb{N}$. Wenn $x = 0$, so ist $t_0 = a^n$ für ein $n \in \mathbb{N}$. Gib n aus. Mögen jetzt $x \geq 1$ und h die zuletzt auf t_{x-1} produzierte Hypothese sein. Ist $t_x^+ \subseteq L_h$, so gib h aus. Andernfalls bestimme die kleinste Zahl z mit $a^{z+1} \notin t_x^+$. Gib z aus.

Nach Konstruktion arbeitet M konservativ und konsistent. Wenn M auf t konvergiert, sagen wir gegen j, so muß $L = L_j$ gelten, da die Sprachen aus $\mathcal{L}_{dmon}$ paarweise unver-

gleichbar sind. Schließlich konvergiert M auf t. Um das zu sehen, sei j die Nummer der zu lernenden Sprache L. Also ist $a^{j+1} \notin L$ und $a^k \in L$ für alle $k \neq j+1$. Für alle hinreichend großen x ergibt sich daraus, daß j die kleinste Zahl z mit $a^{z+1} \notin t_x^+$ ist. Da $t_{x+r}^+ \subseteq L_j$ für alle $r \in \mathbb{N}$, wird die Hypothese j in jedem nachfolgenden Lernschritt wiederholt.

q.e.d.

Wie unser nächster Satz zeigt, kann ein zusätzlich erlaubter Hypothesenwechsel im allgemeinen weder durch eine geeignete Wahl des Hypothesenraumes noch durch mehr Informationen über die zu lernende Sprache kompensiert werden.

Satz 4.7. *$ELIM_{k+1}-TXT \setminus LIM_k-INF \neq \emptyset$ für alle $k \geq 0$.*

Beweis: Sei $k \geq 0$. Mit $\mathcal{L}_{LIM_{k+1}}$ bezeichnen wir die Familie aller Sprachen über dem Alphabet $\Sigma = \{a\}$, die aus höchstens $k+2$ Elementen bestehen, wobei diese Sprachen in kanonischer Weise rekursiv aufgezählt sein mögen. Zum Nachweis von $\mathcal{L}_{LIM_{k+1}} \in ELIM_{k+1}-TXT$ genügt es nun, sich zu überlegen, daß man einfach eine kanonische Nummer von t_x^+ ausgeben kann. Da jede Sprache $L \in \mathcal{L}_{LIM_{k+1}}$ höchstens $k+2$ Elemente enthält, werden höchstens $k+1$ Hypothesenwechsel ausgeführt.

Es bleibt $\mathcal{L}_{LIM_{k+1}} \notin LIM_k-INF$ zu zeigen. Angenommen, es gibt eine IIM M mit $\mathcal{L}_{LIM_{k+1}} \subseteq LIM_k-INF(M)$. Möge $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ ein beliebig fixierter klassentreuer Hypothesenraum sein. Sei i der lexikographisch geordnete Informant von $L = \{a\}$. Da M die Sprache L lernen muß, existiert ein x derart, daß $M(i_x) = j_x$ mit $L = L(G_{j_x})$. Offensichtlich kann i_x jetzt zum lexikographisch geordneten Informanten von $\tilde{L} = \{a, a^{x+1}\}$ fortgesetzt werden. Also hat die Maschine M auf dieser Fortsetzung mindestens einen Hypothesenwechsel auszuführen, da andernfalls $\tilde{L}$ nicht gelernt wird. Damit hat man schon einen Widerspruch für den Fall $k = 0$. Durch Iteration des gleichen Arguments erhält man leicht die Existenz einer Sprache $\hat{L} \in \mathcal{L}_{LIM_{k+1}}$ mit genau $k+2$ Elementen, auf der M mindestens $k+1$ Hypothesenwechsel abverlangt werden, wenn als Eingabe ihr lexikographisch geordnete Informant vorliegt.

q.e.d.

Damit haben wir die folgende Hierarchie erhalten.

$$LIM_0-TXT \subset ... \subset LIM_k-TXT \subset ... \subset \bigcup_{k \in \mathbb{N}} LIM_k-TXT \subset LIM-TXT$$

Es ist nicht sonderlich schwer, $FIN-INF \subset$ *ECONSERVATIVE–TXT* zu beweisen. Wir verweisen dazu auf Satz 4.29, wo das wesentlich schärfere Resultat $ESMON–INF \subset$ *ECONSERVATIVE–TXT* gezeigt wird. Erstaunlicherweise ändert sich die Situation bereits bei einem erlaubten Hypothesenwechsel vollständig. Es sei noch angemerkt, daß dieser Wechsel, nach dem gerade zitierten Satz, offenbar nicht streng-monoton sein kann.

Satz 4.8. $ELIM_1{-}INF \setminus LIM{-}TXT \neq \emptyset$

Beweis: Wir definieren $\mathcal{L} = L_0, L_1, ...$ über dem Alphabet $\{a\}$ wie folgt: $L_0 = \{a\}^*$, $L_k = \{a^n \mid 1 \leq n \leq k\}$ für alle $k \geq 1$. Analog wie im Beweis von Satz 4.3 überzeugt man sich leicht davon, daß $\mathcal{L} \notin LIM{-}TXT$. Zum Nachweis von $\mathcal{L} \in ELIM_1{-}INF$ definieren wir informal eine IIM M. Seien $L \in \mathcal{L}$ und $i \in \mathit{Info}(L)$. Solange kein Paar der Form $(a^n, -)$ gelesen wird, gibt M die Hypothese 0 aus. Wird ein Paar $(a^n, -)$ gelesen, so wiederholt M ihre letzte Hypothese solange, bis beide Paare $(a^k, +)$ and $(a^{k+1}, -)$ gelesen wurden. Danach wird in allen folgenden Schritten k als Hypothese ausgegeben. Aus der Konstruktion der indizierten Familie $\mathcal{L}$ und der Definition von M sieht man nun sofort $\mathcal{L} \subseteq ELIM_1{-}INF(M)$.

q.e.d.

Da nun weiterhin natürlich $LIM_k{-}TXT \subseteq LIM_k{-}INF$ und nach Satz 4.3 auch $FIN{-}TXT \subset FIN{-}INF$ gilt, erlauben die Sätze 4.7 und 4.8 folgende Schlußfolgerungen.

Corollar 4. 9.

(1) $LIM_k{-}TXT \subset LIM_k{-}INF$ *für alle* $k \geq 0$.

(2) $LIM_k{-}INF \subset LIM_{k+1}{-}INF$ *für alle* $k \geq 0$.

(3) $\bigcup_{k \in \mathbb{N}} LIM_k{-}INF \subset LIM{-}INF$.

(4) $LIM{-}TXT \# \bigcup_{k \in \mathbb{N}} LIM_k{-}INF$.

Beweis: (1) und (2) sind nach den vorangegangenen Bemerkungen klar. Zum Beweis von (3) genügt der Nachweis von $LIM{-}INF \setminus \bigcup_{k \in \mathbb{N}} LIM_k{-}INF \neq \emptyset$, da die einfache Inklusion sich direkt aus Definition 14 ergibt. Mit $\mathcal{L}_{fin}$ bezeichnen wir die indizierte Familie aller endlichen Sprachen über dem Alphabet $\{a\}$ in kanonischer Aufzählung. Offensichtlich gilt $\mathcal{L}_{fin} \in LIM{-}INF$. Mit Standardargumenten überlegt man sich leicht, daß es kein k mit $\mathcal{L}_{fin} \in LIM_k{-}INF$ geben kann.

Zum Beweis von (4) stellen wir zunächst fest, daß offenbar auch $\mathcal{L}_{fin} \in LIM{-}TXT$ gilt. Nach Satz 4.8 erhält man weiterhin sofort $\bigcup_{k \in \mathbb{N}} LIM_k{-}INF \setminus LIM{-}TXT \neq \emptyset$, woraus (4) folgt.

q.e.d.

Zusammenfassend erhalten wir nachstehende Hierarchie.

$$LIM_0{-}INF \subset LIM_1{-}INF ... \subset LIM_k{-}INF \subset ... \subset \bigcup_{k \in \mathbb{N}} LIM_k{-}INF \subset LIM{-}INF$$

Wir kommen jetzt zum interessantesten Teil dieses Abschnittes. Unser Ziel besteht darin, exaktes und klassentreues Lernen unter einer a priori begrenzten Anzahl von erlaubten Hypothesenwechseln zu vergleichen. Von speziellem Interesse ist dabei natürlich auch das Verhältnis zu konservativem Lernen. Analysiert man Angluins (1980a) Beweis des Satzes $ECONSERVATIVE\text{–}TXT \subset ELIM\text{–}TXT$ genauer, so sieht man, daß bereits drei beliebige Hypothesenwechsel genügen, um Klassen inferieren zu können, die keine exakt lernende konservative IIM erkennt. Wir wollten es genauer wissen. Dazu haben wir eine neue Beweistechnik entwickelt. Weitere Anwendungen dieser Technik stellen wir im nächsten Abschnitt vor.

Satz 4.10.

(1) $ELIM_1\text{–}TXT \subset ECONSERVATIVE\text{–}TXT$

(2) $LIM_1\text{–}TXT \subset CONSERVATIVE\text{–}TXT$

(3) $LIM_1\text{–}TXT \setminus ECONSERVATIVE\text{–}TXT \neq \emptyset$

Beweis: (1) und (2) werden völlig analog bewiesen. Wir zeigen deshalb hier nur (1). Sei $\mathcal{L}$ eine beliebig fixierte indizierte Familie mit $\mathcal{L} \in ELIM_1\text{–}TXT$. Dann existiert eine IIM M, die $\mathcal{L}$ bei Textbelehrung mit höchstens einem Hypothesenwechsel lernt. Seien $L \in \mathcal{L}$ und $t \in Text(L)$. Dann gilt für alle $x \in \mathbb{N}$: Wenn $j_x = M(t_x)$, so $L \not\subset L_{j_x}$. Um das zu sehen, nehmen wir an, es gäbe ein x mit $L \subset L_{j_x}$. Dann kann j_x nicht korrekt für L sein. Damit darf M bis zur Ausgabe von j_x noch keinen Hypothesenwechsel ausgeführt haben. Folglich muß ein $r \in \mathbb{N}$ derart exisitieren, daß $j_{x+r} = M(t_{x+r})$ und $L = L_{j_{x+r}}$. Da aber $L \subset L_{j_x}$ ist, erhalten wir andererseits sofort $t^+_{x+r} \subset L_{j_x}$. Also ist t_{x+r} auch Anfangstück eines Textes $\tilde{t}$ für L_{j_x}. Wegen $L = L_{j_{x+r}}$ ist j_{x+r} aber keine korrekte Hypothese für L_{j_x}. Damit wechselt M aber auf $\tilde{t}$ mindestens zweimal die Hypothese oder L_{j_x} wird gar nicht bei Belehrung durch $\tilde{t}$ gelernt. In beiden Fällen ergibt sich sofort ein Widerspruch zu $\mathcal{L} \subseteq ELIM_1\text{–}TXT(M)$. Folglich produziert M auf allen Texten für Sprachen aus $\mathcal{L}$ nie eine übergeneralisierte Hypothese. Somit macht es nun keine Mühe, M in eine konservativ arbeitende IIM zu transformieren. Damit ist (1) gezeigt.

Der Beweis von Aussage (3) ist deutlich schwerer. Möge $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die Cantorfunktion bezeichnen. Wir definieren die gewünschte indizierte Familie wie folgt. Für alle $k, j \in \mathbb{N}$ sei

$L_{c(k,0)} = \{a^k b^n \mid n \in \mathbb{N}^+\}$ und für $j \geq 1$ sei

$$L_{c(k,j)} = \begin{cases} L_{c(k,0)}, & \text{falls } \neg\Phi_k(k) \leq j, \\ \{a^k b^m \mid 1 \leq m \leq \Phi_k(k)\}, & \text{sonst.} \end{cases}$$

Nach Konstruktion ist $\mathcal{L}$ eine indizierte Familie, da das Prädikat „$\Phi_i(y) \leq z$“ nach der Definition von Komplexitätsmaßen uniform für alle $i, y, z \in \mathbb{N}$ entscheidbar ist (cf., Blum, 1967). Wir beweisen zuerst $\mathcal{L} \notin$ *ECONSERVATIVE–TXT*. Konzeptionell verfahren wir dabei folgendermaßen. Wir geben eine Methode an, mit deren Hilfe sich aus jeder IIM, die $\mathcal{L}$ exakt konservativ lernt, effektiv ein Algorithmus zur Lösung des Halteproblems gewinnen läßt. Da das Halteproblem algorithmisch unlösbar ist, folgt somit $\mathcal{L} \not\subseteq$ *ECONSERVATIVE–TXT*(M) für alle IIMs M.

***Hilfssatz* A**. Wenn $\mathcal{L} \subseteq$ *ECONSERVATIVE–TXT*(M) für eine IIM M, so kann „$\varphi_k(k)\downarrow$“ effektiv entschieden werden.

Möge $\mathcal{L} \subseteq$ *ECONSERVATIVE–TXT*(M) für eine IIM M sein. Wir beschreiben einen Algorithmus zur Lösung des Halteproblems. Bei Eingabe von k arbeitet der Algorithmus folgende Anweisungen ab:

(A1) Generiere sukzessive den kanonischen Text t von $L_{c(k,0)}$ bis M nach dem Lesen von t_z die erste Hypothese j_z mit $t_z^+ \subseteq L_{j_z}$ ausgibt.
(* Da k fixiert ist, muß j_z von der Form $c(k,i)$ sein. *)

(A2) Teste, ob $a^k b^{x+1} \in L_{c(k,x)}$, wobei $x = max\{i, z\}$. Wenn nicht, gib „$\varphi_k(k)$ definiert“ aus. Andernfalls gib „$\varphi_k(k)$ nicht definiert“ aus.

Wir haben noch zu zeigen, daß das eben beschriebene Verfahren ein Algorithmus ist und das Verlangte leistet. Dazu erinnern wir zunächst daran, daß der kanonische Text jeder Sprache einer indizierten Familie rekursiv generierbar ist. Die uniforme Entscheidbarkeit des Wortproblems für alle Sprachen aus $\mathcal{L}$ sichert weiterhin die effektive Ausführbarkeit der Anweisungen (A1) und (A2). Die Abarbeitung von (A1) muß terminieren, da andernfalls $L_{c(k,0)}$ von M nicht gelernt wird. Somit terminiert der Algorithmus bei jeder Eingabe.

Wir zeigen jetzt die Korrektheit. Ist nun $a^k b^{x+1} \notin L_{c(k,x)}$, so ist $L_{c(k,x)}$ nach Konstruktion endlich. Aus der Endlichkeit von $L_{c(k,x)}$ folgt aber unmittelbar $\Phi_k(k) \leq n$ für ein n, also ist $\varphi_k(k)$ definiert.

Möge $a^k b^{x+1} \in L_{c(k,x)}$ sein. Dann ist $L_{c(k,x)}$ unendlich, da andernfalls nach Konstruktion $\Phi_k(k) \leq x$ wäre, woraus nach der Definition von $\mathcal{L}$ sofort $a^k b^{x+1} \notin L_{c(k,x)}$ folgte.

Also gilt $\neg\Phi_k(k) \leq x$ und $L_{c(k,x)} = L_{c(k,0)}$. Wir haben zu zeigen, daß $\varphi_k(k)$ nicht definiert ist. Angenommen, $\varphi_k(k)$ wäre definiert. Dann muß nach Konstruktion ein $m > x$ derart existieren, daß $L_{c(k,m)}$ endlich ist. Da $m > x \geq max\{i, z\}$ ist, ergibt sich unmittelbar, daß t_z auch Anfangsstück des kanonischen Textes von $L_{c(k,m)}$ ist. Damit hat die Maschine M aber auf diesem Anfangsstück t_z des kanonischen Textes von $L_{c(k,m)}$ eine übergeneralisierte Hypothese (nämlich $c(k,i)$) ausgegeben. Somit arbeitet M nicht konservativ bzgl. des Hypothesenraumes $\mathcal{L}$. Damit ist die Korrektheit des Algorithmus gezeigt.

q.e.d. (Hilfssatz A)

Wir beenden den Beweis von Aussage (3) mit dem folgenden Hilfssatz.

***Hilfssatz* B.** $\mathcal{L} \in LIM_1 - TXT$

Wir definieren zunächst eine indizierte Familie $\hat{\mathcal{L}} = (\hat{L}_j)_{j\in\mathbb{N}}$ als Hypothesenraum. Sei $k \in \mathbb{N}^+$. Wir setzen:

$$\hat{L}_{2k-2} = L_{c(k,0)}$$

$$\hat{L}_{2k-1} = \bigcap_{j\in\mathbb{N}^+} L_{c(k,j)}$$

Aus der Definition erhalten wir sofort, daß $\hat{\mathcal{L}}$ klassentreu ist. Wir haben noch die uniforme Entscheidbarkeit des Wortproblems zu zeigen. Für gerade Indizes ist das unmittelbar klar. Sei jetzt $z = 2k-1$ und w ein beliebiges Wort über $\{a, b\}$. Dann gilt: $w \in \hat{L}_{2k-1}$ genau dann, wenn es ein j mit $w = a^k b^j$ und $w \in L_{c(k,j)}$ gibt. Diese Äquivalenz ergibt sich unmittelbar aus der Definition von $\mathcal{L}$, denn nach Konstruktion ist $a^k b^j \in L_{c(k,j)}$ genau dann, wenn $a^k b^j \in L_{c(k,j+r)}$ für alle $r \in \mathbb{N}$. Also ist das Wortproblem insgesamt uniform entscheidbar. Somit ist $\hat{\mathcal{L}}$ ein klassentreuer Hypothesenraum.

Wir definieren jetzt eine IIM M, die $\mathcal{L}$ bzgl. $\hat{\mathcal{L}}$ mit jeweils höchstens einem Hypothesenwechsel lernt. Seien $L \in \mathcal{L}$, $t \in Text(L)$ und $x \in \mathbb{N}$. Wir setzen:

$$M(t_x) = \begin{cases} 2k-1, & \text{falls } t_x^+ \subseteq \hat{L}_{2k-1}, \\ 2k-2, & \text{sonst.} \end{cases}$$

Wir haben noch zu zeigen, daß M das Verlangte leistet. Nach der Definition der Familien $\mathcal{L}$ und $\hat{\mathcal{L}}$ ist klar, daß M höchstens einen Hypothesenwechsel ausführt. Ist die zu lernende Sprache L endlich, so gibt M in jedem Schritt die gleiche korrekte Hypothese aus. Wenn L unendlich ist, kann M endlich oft dieselbe falsche Hypothese j ausgeben. Dann ist aber $\hat{L}_j$ endlich. Somit muß ein y mit $t_y^+ \not\subseteq \hat{L}_j$ existieren. Nach Definition von M führt M also spätestens nach dem Lesen von t_y einen Hypothesenwechsel aus und produziert anschließend nur korrekte Hypothesen. q.e.d. (Hilfssatz B)

Aus den Hilfssätzen A und B folgt direkt Aussage (3) des Satzes.

q.e.d.

Das nachstehende Corollar ist eine unmittelbare Folgerung aus dem letzten Satz.

Corollar 4. 11.

(1) $ECONSERVATIVE\text{–}TXT \subset CONSERVATIVE\text{–}TXT$

(2) $ELIM_1\text{–}TXT \subset LIM_1\text{–}TXT$

Die im Beweis des letzten Satzes eingeführte Beweistechnik kann nun so modifiziert werden, daß wir mit ihrer Hilfe das Verhältnis von exaktem und klassentreuen Lernen charakterisieren können.

Satz 4.12. *Für alle $k \geq 1$ gilt: $ELIM_k\text{–}TXT \subset LIM_k\text{–}TXT$*

Beweis: Wir separieren zunächst $ELIM_2\text{–}TXT$ und $LIM_2\text{–}TXT$ und erklären hierbei die Beweisidee. Anschließend verallgemeinern wir den Beweis induktiv. Sei $\mathcal{L}$ die im Beweis von Satz 4.10 eingeführte indizierte Familie. Wir definieren für jede Sprache $L_m \in \mathcal{L},\ m \in \mathbb{N}$ eine Sprache $\tilde{L}_m = L_m \cup \{b\}$. Weiterhin möge $\mathcal{L}_{sep2}$ die kanonische Aufzählung aller Sprachen L_m und $\tilde{L}_m$ sein. Wie im Beweis von Satz 4.10 definieren wir mutatis mutandis einen klassentreuen Hypothesenraum $\hat{\mathcal{L}}$ für $\mathcal{L}_{sep2}$. Mit den gleichen Ideen wie im Beweis von Hilfsatz B des Satzes 4.10 zeigt man mühelos $\mathcal{L}_{sep2} \in LIM_2\text{–}TXT$ bzgl. $\hat{\mathcal{L}}$. Es bleibt also noch $\mathcal{L}_{sep2} \notin ELIM_2\text{–}TXT$ nachzuweisen. Wir führen den Beweis indirekt mit Hilfe des folgenden Hilfssatzes.

***Hilfssatz* A**. Wenn $\mathcal{L}_{sep2} \in ELIM_2\text{–}TXT$, so ist $\mathcal{L} \in ELIM_1\text{–}TXT$.

Möge M eine beliebig fixierte IIM mit $\mathcal{L}_{sep2} \subseteq ELIM_2\text{–}TXT(M)$ sein. Weiterhin seien $\tilde{L}_m = L_m \cup \{b\}$ und $t \in \mathit{Text}(L_m)$ beliebig fixiert. Da M auch die Sprache L_m lernen muß, existiert ein $z \in \mathbb{N}$, so daß $M(t_x) = j$ für alle $x \geq z$ und $L_m = L_j$. Wir betrachten jetzt das Verhalten von M auf dem folgenden Text $\tilde{t}$ für $\tilde{L}_m$. Der Text $\tilde{t}$ beginnt mit t_z, wird dann mit dem Wort b fortgesetzt und schließlich mit dem kanonischen Text von $\tilde{L}_m$ zu einem Text vervollständigt. Nach Konstruktion ist $M(t_z) = j$ und $\tilde{L}_m \neq L_j$. Folglich muß M auf $\tilde{t}$ noch mindestens einen Hypothesenwechsel ausführen. Da M aber auf jedem Text einer zu $\mathcal{L}_{sep2}$ gehörenden Sprache höchstens zwei Hypothesenwechsel ausführen darf, kann M auf dem Anfangsstück t_z die Hypothese höchstens einmal gewechselt haben. Also ist $\mathcal{L} \in ELIM_1\text{–}TXT(M)$.

q.e.d. (Hilfssatz A)

Damit ist der Satz für $k = 2$ gezeigt.

Der Rest wird induktiv bewiesen. Als Induktionsvoraussetzung nehmen wir an, daß

wir bereits $ELIM_k{-}TXT$ und $LIM_k{-}TXT$ unter Verwendung der indizierten Familie $\mathcal{L}_{sepk} = (L_m)_{m\in\mathbb{N}}$ separiert haben. Wir setzen jetzt für jedes $m \in \mathbb{N}$ die Sprache $\tilde{L}_m = L_m \cup \{b, b^2, ..., b^k\}$ und definieren $\mathcal{L}_{sep(k+1)}$ als kanonische Aufzählung aller Sprachen aus $\mathcal{L}_{sepk}$ und aller Sprachen $\tilde{L}_m$. Mutatis mutandis definiert man wieder wie im Beweis von Satz 4.10 einen klassentreuen Hypothesenraum $\hat{\mathcal{L}}$ und zeigt $\mathcal{L}_{sep(k+1)} \in LIM_{k+1}{-}TXT$ bzgl. $\hat{\mathcal{L}}$. Eine leichte Modifikation des oben gegebenen Beweises von Hilfssatz A liefert dann unmittelbar folgenden Hilfssatz B.

***Hilfssatz* B**. Wenn $\mathcal{L}_{sep(k+1)} \in ELIM_{k+1}{-}TXT$, so ist $\mathcal{L}_{sepk} \in ELIM_k{-}TXT$.

Hilfssatz B liefert somit einen Widerspruch zur Induktionsvoraussetzung, womit $\mathcal{L}_{sep(k+1)} \notin ELIM_{k+1}{-}TXT$ gezeigt ist.

q.e.d.

An dieser Stelle sind noch einige Bemerkungen angezeigt. Es blieb offen, ob Satz 4.10 zu $LIM_k{-}TXT \setminus CONSERVATIVE{-}TXT \neq \emptyset$ verschärft werden kann. Wir konnten jedoch eine partielle Antwort finden, die auch für sich selbst genommen, von Interesse ist. Inhaltlich geht es darum, ob der Lernprozeß stets durch IIMs realisiert werden kann, die *semantisch finit* arbeiten. Wir sagen von einer IIM M, daß sie semantisch finit arbeitet, wenn für alle $L \in \mathcal{L}$, jeden Text $t \in Text(L)$ folgendes gilt: Sei j diejenige Hypothese, gegen die $(M(t_x))_{x\in\mathbb{N}})$ konvergiert und z die kleinste Zahl, für die $M(t_z) = j$ ist. Dann ist $L(G_{M(t_y)}) \neq L(G_j)$ für alle $y < z$. Eine semantisch finit arbeitende IIM konvergiert also gegen die erste korrekte Hypothese in ihrer Ausgabefolge. Interessanterweise kann man jede im Sinne von EX lernbare Funktionenklasse auch semantisch finit inferieren (cf., Wiehagen, 1991). Wie sieht dies nun für Sprachen aus? Aus der Definition von Konservativität ergibt sich unmittelbar die semantische Finitheit konservativer IIMs. Aber es ist zunächst völlig offen, ob konservatives Arbeiten die einzige Form ist, semantische Finitheit zu realisieren. Mit den folgenden Sätzen beantworten wir diese Frage für exaktes und klassentreues Lernen.

Satz 4.13. *Seien $\mathcal{L}$ eine indizierte Familie und $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum mit $\mathcal{L} = \{L(G_j) | j \in \mathbb{N}\}$. Wenn es eine semantisch finit arbeitende IIM gibt, die $\mathcal{L}$ bei Textbelehrung im Limes bzgl. $\mathcal{G}$ (bzgl. $\mathcal{L}$) lernt, so ist $\mathcal{L} \in CONSERVATIVE{-}TXT$ ($\mathcal{L} \in ECONSERVATIVE{-}TXT$).*

Beweis: Der klassentreue und der exakte Fall können analog bewiesen werden. Deshalb zeigen wir hier nur die Behauptung für den klassentreuen Fall. Möge $\mathcal{L}$ von der IIM M semantisch finit bzgl. $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ gelernt werden. Wir beginnen mit folgendem Hilfssatz.

Hilfssatz. Seien $L \in \mathcal{L}$ und $t \in Text(L)$. Dann produziert M auf t nie eine Hypothese

n mit $L \subset L(G_n)$.

Angenommen doch; dann existiert ein $x \in \mathbb{N}$ derart, daß $M(t_x) = n$ und $L \subset L(G_n)$. Da $L \in \mathcal{L}$ ist und folglich von M gelernt werden muß, gibt es somit ein $r \in \mathbb{N}^+$ mit $M(t_{x+r}) = m$ und $L = L(G_m)$. Aus $L \subset L(G_n)$ erhalten wir weiterhin, daß t_{x+r} auch Anfangsstück eines Textes $\tilde{t}$ für $L(G_n) \in \mathcal{L}$ ist. Nach Konstruktion gilt aber $M(\tilde{t_x}) = n$, jedoch $M(\tilde{t_{x+r}}) = m$ und $L(G_m) \neq L(G_n)$. Also arbeitet M auf $\tilde{t}$ nicht semantisch finit.

q.e.d. (Hilfssatz)

Wir definieren jetzt eine IIM $\hat{M}$ wie folgt: Seien $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$.

$\hat{M}(t_x) =$ „Wenn $x = 0$ oder M nach dem sukzessiven Lesen von t_{x-1} noch keine Hypothese produziert hat, so gehe zu (A). Andernfalls arbeite (B) ab.

(A) Simuliere M bis zum Lesen von t_x. Gibt M nach dem Lesen von t_x eine Hypothese, sagen wir j, aus, so gib j aus. Andernfalls gib keine Hypothese aus und lies t_{x+1}.

(B) Möge h die letzte nach dem Lesen von t_{x-1} produzierte Hypothese von $\hat{M}$ sein. Teste, ob $t_x^+ \subseteq L(G_h)$. Wenn ja, so gib h aus. Andernfalls arbeite (A) ab.“

Nach Konstruktion arbeitet $\hat{M}$ offenbar konservativ. Der obige Hilfssatz liefert weiterhin die Aussage, daß jede Hypothese, gegen die $\hat{M}$ konvergiert, auch korrekt für die betreffende Sprache ist. Da die IIM M nach Voraussetzung jede Sprache $L \in \mathcal{L}$ lernt, konvergiert nach Konstruktion auch $\hat{M}$ auf jedem Text einer Sprache $L \in \mathcal{L}$.

q.e.d.

Aus dem Beweis des letzen Satzes läßt sich noch folgendes Corollar ableiten.

Corollar 4. 14. *Wenn $LIM_k{-}TXT \setminus$ CONSERVATIVE–TXT $\neq \emptyset$, so gibt es eine indizierte Familie $\mathcal{L} \in LIM_k{-}TXT$, die nicht semantisch finit erlernbar ist.*

Beweis: Die im Beweis des letzten Satzes konstruierte IIM $\hat{M}$ wechselt auf jedem Text einer Sprache $L \in \mathcal{L}$ sicher nicht öfter ihre Hypothese als die dort zugrunde gelegte semantisch finit arbeitende IIM M. Somit ist jede mit höchstens k Hypothesenwechseln semantisch finit erlernbare indizierte Familie auch mit höchstens k Hypothesenwechseln konservativ erlernbar. Die Kontraposition dieser Beobachtung ergibt die Behauptung des Corollars.

q.e.d.

Schließlich lassen sich die letzen beiden Sätze noch zu folgender Charakterisierung von *CONSERVATIVE–TXT* zusammenfassen.

Corollar 4. 15. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in$ CONSERVATIVE–TXT genau dann, wenn $\mathcal{L}$ im Limes semantisch finit bzgl. eines Hypothesenraumes $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ mit $\mathcal{L} = \{L(G_j) | j \in \mathbb{N}\}$ gelernt werden kann.*

Beweis: Die Hinlänglichkeit ist in Satz 4.13 gezeigt worden. Zum Beweis der Notwendigkeit genügt es, sich in Erinnerung zu rufen, daß eine konservativ arbeitende IIM nur durch Inkonsistenz verursachte Hypothesenwechsel ausführen darf. Damit konvergiert aber jede konservative IIM gegen die erste korrekte Hypothese.

q.e.d.

Das folgende Bild faßt die in diesem Abschnitt erhaltenen Resultate zusammen. Dabei hat eine durchgezogene Linie zwischen zwei Lerntypen immer die Bedeutung, daß der obere Lerntyp den unteren echt umfaßt. Um die Aussagekraft des Bildes nicht zu mindern, haben wir die Linie zwischen $FIN-INF$ und *CONSERVATIVE–TXT* weggelassen.

Lernen unter einer begrenzten Anzahl von Hypothesenwechseln und konservative Inferenz

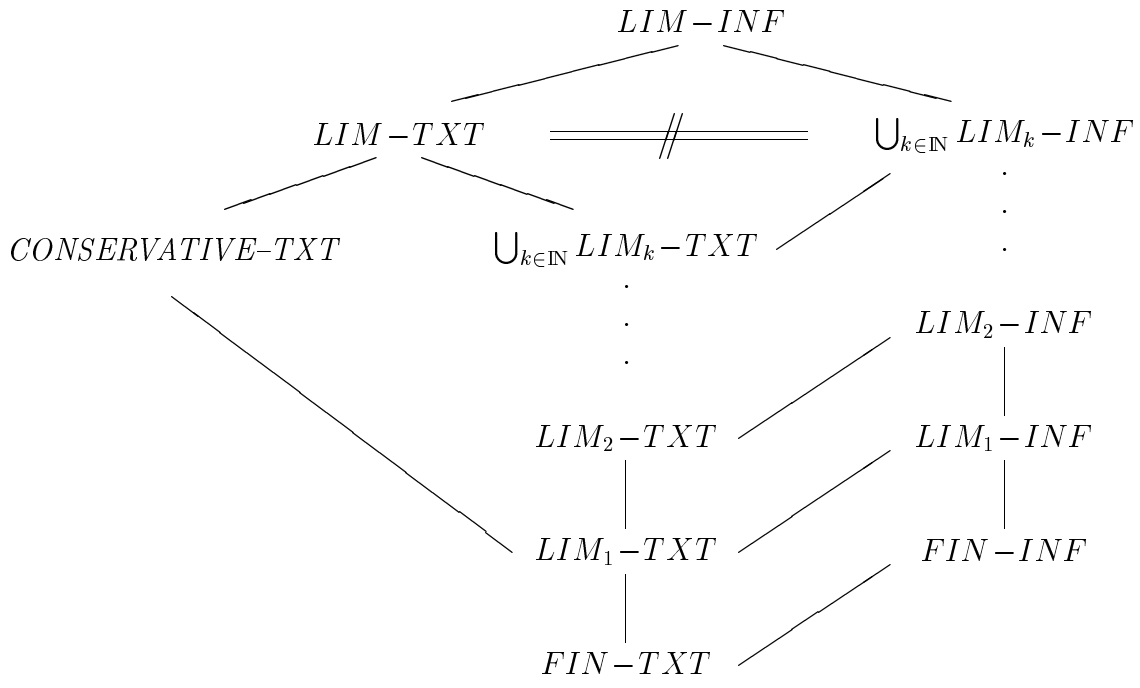

Abbildung 8

4.1.3. Monotones und dual monotones Lernen

In diesem Abschnitt studieren wir den Einfluß der in Abschnitt 2.3 definierten Monotonieforderungen auf die Leistungsfähigkeit von Lernalgorithmen. Lernen unter Monotonieforderungen wurden erstmals in Jantke (1991a, 1991b) studiert. Freivalds, Kinber und Wiehagen (1992) erlaubt für die Inferenz rekursiver Funktionen wesentlich vertiefte Einsichten in die Arbeitsweise streng-monotoner induktiver Inferenzmaschinen. Das systematische Studium des Lernens indizierter Familien rekursiver Sprachen durch Generalisieren haben wir in Lange und Zeugmann (1992a, 1992b) begonnen. Nachdem die ersten Resultate vorlagen führte Kapur (1992) die dualen Monotoniebegriffe ein. Für eine ausführliche Darstellung der erzielten Resultate im klassentreuen Fall verweisen wir auf Lange, Zeugmann und Kapur (1992). In der Folge wurde von S. Lange gemeinsam mit dem Autor entdeckt, daß der Wahl des Hypothesenraumes eine entscheidene Bedeutung zukommt, sofern im Lernprozeß Monotonieforderungen eingehalten werden müssen (cf., Lange und Zeugmann, 1993c). Allerdings kann damit die Strenge der Forderungen, monoton bzw. dual monoton lernen zu müssen nur gemildert, jedoch nicht aufgehoben werden. Das zeigt uns der erste Satz dieses Abschnittes.

Satz 4.16.

(1) $EMON\text{-}TXT \setminus KSMON\text{-}INF \neq \emptyset$

(2) $EWMON\text{-}TXT \setminus KMON\text{-}INF \neq \emptyset$

(3) $EMON^d\text{-}TXT \setminus KSMON^d\text{-}INF \neq \emptyset$

(4) $EWMON^d\text{-}TXT \setminus KMON^d\text{-}INF \neq \emptyset$

Beweis: Wir zeigen zunächst Aussage (1) und (3). Dazu verwenden wir folgende indizierte Familie. Sei $(i_0, ..., i_k)$ ein beliebiges $k+1$ Tupel paarweise verschiedener Zahlen aus $\mathbb{N}^+$. Wir setzen $L_\emptyset =_{df} \{a\}^*$ definieren $L_{i_0,...,i_k} =_{df} (\{a\}^* \setminus \{a^{i_0}, ..., a^{i_k}\}) \cup \{b^{i_0}, ..., b^{i_k}\}$. Möge nun $\mathcal{L}_{mon}$ die Familie aller Sprachen $L_\emptyset$, $L_{i_0,...,i_k}$, mit $(i_0, ..., i_k) \in (\mathbb{N}^+)^{k+1}$, $k \in \mathbb{N}^+$, $i_j \neq i_l$ falls $j \neq l$ für alle $j, l \in \{0, ..., k\}$ in kanonischer Aufzählung sein.

Wir geben jetzt eine IIM M an, die $\mathcal{L}_{mon}$ sowohl monoton als auch dual monoton bzgl. $\mathcal{L}_{mon}$ lernt. Dazu seien $L \in \mathcal{L}_{mon}$ und $t \in Text(L)$ beliebig fixiert. Solange der von M gelesene Teil des Textes nur Wörter über dem Alphabet $\{a\}$ enthält, gibt M in jedem Takt die kanonische Nummer von $L_\emptyset$ aus. Wird das erstemal ein Wort b^{l_0} von M gelesen, so gibt M die kanonische Nummer von L_{l_0} aus. Diese Hypothese wird wiederholt, es sei denn, M liest ein weiteres Wort b^{l_1} mit $l_0 \neq l_1$. In diesem Falle ändert M seine

Hypothese in die kanonische Nummer von L_{l_0,l_1} ab. Diese Konstruktion wird iteriert, d.h. M bleibt entweder bei der jeweils aktuellen Hypothese oder liest ein neues, bislang noch nicht angebotene Wort über $\{b\}$. In diesem Fall wird eine kanonische Nummer der neuen Sprache ausgegeben.

Wir haben jetzt zu zeigen, daß M die Familie $\mathcal{L}_{mon}$ tatsächlich lernt und nur Hypothesenwechsel ausführt, die sowohl monoton als auch dual monoton sind. Sei also $L \in \mathcal{L}_{mon}$. Ist $L = \{a\}^*$, so lernt M trivialerweise L. Möge deshalb jetzt $L \neq L_\emptyset$ gelten. Da $L \in \mathcal{L}_{mon}$ existiert mindestens ein Tupel $(i_0, ..., i_k)$ paarweise verschiedener Zahlen, so daß $L = L_{i_0,...,i_k}$. Nach Konstruktion gibt M auf jedem Text $t \in Text(L)$ die kanonische Nummer von $L_{l_0,...,l_k}$ aus, nachdem $b^{l_0}, b^{l_1}, ..., b^{l_k}$ in dieser Reihenfolge von M gelesen wurden. Dabei ist $(l_0, ..., l_k)$ eine Permutation von $(i_0, ..., i_k)$ und folglich gilt $L_{l_0,...,l_k} = L_{i_0,...,i_k}$. Also wird L von M gelernt. Weiterhin führt M insgesamt höchstens $k+1$ Hypothesenwechsel durch. Somit haben wir zum Nachweis der Monotonie die folgende Durchschnitte zu berechnen: $\{a\}^* \cap L = L_{i_0,...,i_k} \setminus \{b^{i_0}, ..., b^{i_k}\}$, $L_{l_0} \cap L = L_{i_0,...,i_k} \setminus \{b^{i_0}, ..., b^{i_k} \mid i_j \neq l_0,\ 0 \leq j \leq k\}$, ... $L_{l_0,...,l_k} \cap L = L_{i_0,...,i_k}$. Sind nun $j_0, ..., j_{k+1}$ die von M produzierten Ausgaben, so gilt offenbar $L_{j_0} \cap L \subseteq L_{j_1} \cap L \subseteq ... \subseteq L_{j_{k+1}} \cap L$, d.h. M arbeitet monoton.

Wie man leicht nachrechnet gilt nun auch $\{a\}^* \cup L \supseteq L_{l_0} \cup L \supseteq ... \supseteq L_{l_0,...,l_k} \cup L = L$. Diese Inklusionen sind aber äquivalent zu $co-(\{a\}^* \cup L) \subseteq co-(L_{l_0} \cup L) \subseteq ... \subseteq co-(L_{l_0,...,l_k} \cup L) = co-L$. Und diese Inklusionen sind äquivalent zu $co-\{a\}^* \cap co-L \subseteq co-L_{l_0} \cap co-L \subseteq ... \subseteq co-L_{l_0,...,l_k} \cap co-L = co-L$. Also arbeitet M auch dual monoton.

Es bleibt jetzt noch zu zeigen, daß $\mathcal{L}_{mon}$ weder streng-monoton noch dual streng-monoton bei Informantbelehrung bzgl. eines klassenumfassenden Hypothesenraumes erlernbar ist. Da der Beweis in beiden Fällen fast gleich ausfällt, zeigen wir nur $\mathcal{L}_{mon} \notin KSMON-INF$ und geben anschließend das zu änderne Detail an.

Angenommen, es gäbe eine IIM M sowie einen Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$, so daß $\mathcal{L}_{mon} \subseteq KSMON-INF(M)$ bzgl. $\mathcal{G}$. Mögen $l_0 \in \mathbb{N}^+$ beliebig fixiert und $i \in Info(L_{l_0})$ sein. Da L_{l_0} von M gelernt wird, muß ein x existieren, so daß M nach dem Lesen von i_x eine für L_{l_0} korrekte Hypothese j_x ausgibt. Sei jetzt $l_1 \in \mathbb{N}^+$ irgendeine Zahl mit $l_1 > max\{|w| \mid w \in Val\ i_x\}$. Wir betrachten dann das Verhalten von M auf i_x und $i_x, (b^{l_1}, +)$, d.h. auf zwei Anfangsstücken für Informanten von L_{l_0,l_1}. Somit muß M auf einer hinreichend langen Fortsetzung von $i_x, (b^{l_1}, +)$ zu einem Informanten für L_{l_0,l_1} mindestens einen Hypothesenwechsel ausführen um eine für L_{l_0,l_1} korrekte Hypothese zu produzieren, da $L(G_{j_x}) \neq L_{l_0,l_1}$. Nun gilt aber $a^{l_1} \in L_{l_0} \setminus L_{l_0,l_1}$ und folglich $L_{l_0} \not\subseteq L_{l_0,l_1}$. Somit verletzt M mindestens einmal die Forderung, streng-monoton zu arbeiten.

Mit den gleichen Argumenten sieht man auch, daß M nicht dual streng-monoton arbeitet. Die einzige Modifikation ist die Feststellung $b^{l_1} \in co{-}L_{l_0} \setminus co{-}L_{l_0,l_1}$. Damit sind (1) und (3) gezeigt.

Zum Beweis von Aussage (2) nutzen wir die indizierte Familie $\mathcal{L}_{wmon}$ definiert als $L_0 =_{df} \{a\}^*$ und für $i \geq 1$ seien $L_i =_{df} \{a^z \mid 0 < z \leq i\} \cup \{b^z \mid z > i\}$ und $L_{i,j} =_{df} \{a^z \mid 0 < z \leq i\} \cup \{b^z \mid i < z \leq j\} \cup \{a^z \mid z > j\}$ für alle $i, j \in \mathbb{N}^+$ mit $i < j$. $\mathcal{L}_{wmon}$ ist dann als kanonische Aufzählung aller L_i und $L_{i,j}$ definiert. Wir setzen mit dem Beweis nachstehender Hilfssätze fort.

***Hilfssatz* 1.** $\mathcal{L}_{wmon} \in EWMON{-}TXT$

Die gesuchte IIM M definieren wir informal wie folgt: Sei $L \in \mathcal{L}_{wmon}$ und $t \in Text(L)$. Initial gibt M stets 0 aus. Wenn das erste Wort b^z im Text t gelesen wird, so ändert M die Hypothese und gibt die kanonische Nummer von L_{z-1} aus. Diese Hypothese wird solange wiederholt, bis

(α) ein Wort b^k mit $k < z$ im Text t gelesen wird oder

(β) ein Wort a^m im Text t mit $z < m$ gefunden wird.

Tritt (α) zuerst ein, so gibt M die kanonische Nummer von L_{k-1} aus. Sollte hingegen (β) zuerst eintreten, so gibt M die kanonische Nummer von $L_{z-1,m-1}$ aus. Schließlich kann (α) noch eintreten, nachdem zuerst (β) erfüllt war. In diesem Fall gibt M die kanonische Nummer von $L_{k-1,m-1}$ aus.

Nach Konstruktion führt M Hypothesenwechsel dann und nur dann aus, wenn eine Inkonsistenz zu dem neu gelesenen Datum auftritt. Folglich ist $\mathcal{L}_{wmon} \subseteq EWMON{-}TXT(M)$.

q.e.d. (Hilfssatz 1)

***Hilfssatz* 2.** $\mathcal{L}_{wmon} \notin KMON{-}INF$

Angenommen, es gibt eine IIM M und einen Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ mit $\mathcal{L}_{wmon} \subseteq KMON{-}INF(M)$ bzgl. $\mathcal{G}$. Möge i ein beliebig fixierter Informant von $\{a\}^*$ sein. Da $L_0 \in \mathcal{L}_{wmon}$, existiert folglich ein x, so daß $j_x = M(i_x)$ und $L(G_{j_x}) = L_0$. Wir setzen jetzt i_x sukzessive mit $(b^z, +)$ und $(a^z, -)$ fort, wobei $z > y =_{df} max\{|w| \mid w \in i_x^+ \cup i_x^-\}$. Diese Fortsetzungen i_{x+k} sind Anfangsstücke eines Informanten für L_y. Also muß M irgendwann eine Hypothese j_{x+k} ausgeben, die korrekt für L_y ist. Nun macht es keine Mühe i_{x+k} in kanonischer Art und Weise zu einem Informanten i_{fool} für $L_{y,m}$ fortzusetzen, wobei $m =_{df} max\{|w| \mid w \in i_{x+k}^+ \cup i_{x+k}^-\} + 1$. Wie eine einfache Rechnung zeigt, arbeitet die IIM M nicht monoton auf i_{fool}, sofern sie irgendwann eine für $L_{y,m}$ richtige Hypothese ausgibt.

Falls M also monoton arbeitet, so kann M die Familie $\mathcal{L}_{wmon}$ nicht bei Informantbelehrung im Limes lernen.

q.e.d. (Hilfssatz 2)

Damit haben wir Aussage (2) gezeigt. Der noch ausstehende Beweis von (4) wird mit Hilfe der wie folgt definierten indizierten Familie $\mathcal{L}$ erbracht. Sei $\Sigma = \{a, b, c\}$. Wir setzen $L_0 =_{df} \{a\}^*$, und für $k \geq 1$ sei $L_k =_{df} \{a^z \mid 1 \leq z \leq k\} \cup \{b^z \mid k < z\}$. Schließlich setzen wir $L_{k,m} =_{df} \{a^z \mid 1 \leq z \leq k\} \cup \{b^z \mid k < z \leq m\} \cup \{c^m\}$ für $m > k \geq 1$ und definieren $\mathcal{L}$ als kanonische Aufzählung aller L_k und L_k, m.

***Hilfssatz* 3.** $\mathcal{L} \in EWMON^d\text{-}TXT$

Die gewünschte IIM M, die $\mathcal{L}$ bzgl. des Hypothesenraumes $\mathcal{L}$ lernt, wird informal wie folgt definiert. Seien $L \in \mathcal{L}$, $t \in Text(L)$ und $x \in \mathbb{N}$. Solange $t_x^+ \subseteq \{a\}^*$ ist, gibt M als Hypothese 0 aus. Wird dagegen ein Wort w über $\{b\}$ im Text gelesen, ohne daß M vorher bereits ein Wort c^m gelesen hat, so berechnet die Maschine $k = max\{z \mid a^z \in t_x^+\}$ und gibt die kanonische Nummer von L_k aus. Wenn M ein Wort c^m im Text liest bzw. schon gelesen hat, so berechnet M auch $k = max\{z \mid a^z \in t_x^+\}$ und gibt die kanonische Nummer von $L_{k,m}$ aus.

Man kann sich leicht davon überzeugen, daß M sogar $\mathcal{L}$ dual monoton bzgl. $\mathcal{L}$ lernt.

q.e.d. (Hilfssatz 3)

Wir beenden den Beweis von (4) durch folgenden Hilfssatz.

***Hilfssatz* 4.** $\mathcal{L} \notin KMON^d\text{-}INF$.

Wir nehmen wieder das Gegenteil an. Mögen also $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ ein Hypothesenraum und M eine IIM sein, so daß $\mathcal{L}$ monoton von M bzgl. $\mathcal{G}$ bei Informantbelehrung gelernt wird. Sei i ein Informant für $\{a\}^*$. Dann muß es ein x derart geben, daß für $j_x = M(i_x)$ die Gleichheit $L(G_{j_x}) = \{a\}^*$ gilt. Wir betrachten jetzt $k = max\{|w| \mid w \in i_x^+ \cup i_x^-\}$ und einen beliebig fixierten Informanten $\tilde{i} \in Info(L_k)$ mit $i_x \sqsubseteq \tilde{i}$. Da $L_k \in \mathcal{L}$, muß somit ein $y > x$ existieren, so daß für $j_y = M(\tilde{i}_y)$ auch $L(G_{j_y}) = L_k$ gilt. Wir setzen jetzt $m =_{df} max\{|w| \mid w \in \tilde{i}_y^+ \cup \tilde{i}_y^-\}$. Dann läßt sich $\tilde{i}_y$ als Präfix eines Informanten i_{fool} für $L_{k,m}$ interpretieren. Wird jetzt i_{fool} der Maschine M aufs Band geschrieben, so gibt M irgendwann j_x, danach irgendwann j_y aus und konvergiert schließlich gegen eine korrekte Hypothese für $L_{k,m}$, da M nach Annahme $\mathcal{L}$ lernt. Da jedoch $b^{m+1} \in co\text{-}L_1 \cap co\text{-}L_{k,m}$, aber $b^{m+1} \notin co\text{-}L_k \cap co\text{-}L_{k,m}$ gilt, kann M folglich nicht dual monoton gearbeitet haben.

q.e.d.

Satz 4.16 erlaubt nun folgendes Corollar, das zunächst unsere Begriffsbildung rechtfer-

tigt und die monotonen bzw. dual monotonen Lerntypen hierarchisch ordnet.

Corollar 4. 17.

(1) $ESMON{-}TXT \subset EMON{-}TXT \subset EWMON{-}TXT$

(2) $SMON{-}TXT \subset MON{-}TXT \subset WMON{-}TXT$

(3) $ESMON^d{-}TXT \subset EMON^d{-}TXT \subset EWMON^d{-}TXT$

(4) $SMON^d{-}TXT \subset MON^d{-}TXT \subset WMON^d{-}TXT$

(5) $KSMON^d{-}TXT \subset KMON^d{-}TXT \subset KWMON^d{-}TXT$

Beweis: Aussage (1) und (2) können wieder völlig analog gezeigt werden. Wir beweisen deshalb hier nur (2). Nach Definition 17 gilt offenbar $SMON{-}TXT \subseteq MON{-}TXT$. Satz 4.16 liefert $MON{-}TXT \setminus SMON{-}TXT \neq \emptyset$, womit die erste Inklusion gezeigt ist. Nach Satz 4.16 ist weiterhin $WMON{-}TXT \setminus MON{-}TXT \neq \emptyset$. Es bleibt also $MON{-}TXT \subseteq WMON{-}TXT$ zu zeigen. Inhaltlich besteht das Problem darin, daß eine monotone IIM eventuell eine übergeneralisierte Hypothese ausgeben darf, während dies einer schwach-monoton arbeitenden IIM verwehrt ist. Wie der folgende Hilfssatz zeigt, muß jede übergeneralisierte Hypothese eine *nicht* zur Familie $\mathcal{L}$ gehörende Sprache beschreiben. Somit können im Falle klassentreuer Hypothesenräume keine übergeneralisierten Hypothesen auftreten, woraus sich sofort $MON{-}TXT \subseteq WMON{-}TXT$ ergibt.

Lemma. *Seien $\mathcal{L}$ eine indizierte Familie und $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein für $\mathcal{L}$ klassentreuer Hypothesenraum. Mögen M eine $\mathcal{L}$ bzgl. $\mathcal{G}$ lernende monoton arbeitende IIM, $L \in \mathcal{L}$ und $t \in Text(L)$ sein. Dann gilt: Wenn M auf t eine Hypothese j mit $L \subset L(G_j)$ produziert, so ist $L(G_j) \notin \mathcal{L}$.*

Wir führen den Beweis indirekt. Angenommen, es gibt eine Zahl x, so daß $M(t_x) = j$, $L \subset L(G_j)$ und $L(G_j) \in \mathcal{L}$. Dann ist t_x auch Anfangsstück eines Textes für $L(G_j)$. Andererseits muß es auch eine Zahl k derart geben, daß $t^+_{x+k} \subseteq L$ und $j_{x+k} = M(t_{x+k})$ sowie $L(G_{j_{x+k}}) = L$, da andernfalls L nicht von M gelernt wird. Nun ist aber auch t_{x+k} Anfangsstück eines Textes für $L(G_j)$. Somit läßt sich t_{x+k} zu einem Text $\tilde{t}$ von $L(G_j)$ fortsetzen. Also muß es eine Zahl z derart geben, daß $t_{x+k} \sqsubset \tilde{t}_{x+k+z}$, $\tilde{t}^+_{x+k+z} \subseteq L(G_j)$ und daß für $j_{x+k+z} := M(\tilde{t}_{x+k+z})$ die Gleichheit $L(G_{j_{x+k+z}}) = L(G_j)$ erfüllt ist. Da

$$L(G_j) \supset L = L(G_{j_{x+k}}) \cap L(G_j) \subset L(G_{j_{x+k+z}}) \cap L(G_j) = L(G_{j_{x+k+z}})$$

haben wir aber damit gleichzeitig einen Text $\tilde{t}$ für $L(G_j)$ gefunden, auf dem M nicht monoton arbeitet.

q.e.d. (Lemma)

Zum Nachweis der Aussagen (3), (4) und (5) genügt jetzt ebenfalls der Nachweis der Inklusionen $AMON^d\!-\!TXT \subseteq AWMON^d\!-\!TXT$ für $A \in \{E, \lambda, K\}$, da der Rest unmittelbar aus Definition 18 und Satz 4.16 folgt. Wir zeigen wieder nur den klassentreuen Fall, da die verbleibenden Fälle völlig analog abgehandelt werden können. Mögen also $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein klassentreuer Hypothesenraum und M eine IIM mit $\mathcal{L} \subseteq MON^d\!-\!TXT(M)$ sein. Angenommen, M lernt nicht auch dual schwach-monoton. Dann existieren eine Sprache $L \in \mathcal{L}$ und ein Text $t \in \mathit{Text}(L)$ sowie Zahlen x und r derart, daß $k = M(t_x)$, $j = M(t_{x+r})$, $t^+_{x+r} \subseteq L(G_k)$ und $co\!-\!L(G_k) \not\subseteq co\!-\!L(G_j)$ gilt. Da $t^+_{x+r} \subseteq L(G_k)$ kann das Anfangsstück t_{x+r} zu einem Text $\tilde{t}$ von $L(G_k)$ fortgesetzt werden. Weiterhin muß die Sprache $L(G_k)$ gelernt werden. Also existiert eine Zahl z, so daß $t_{x+r} \sqsubset \tilde{t}_{x+r+z}$ und für die von M auf $\tilde{t}_{x+r+z}$ produzierte Hypothese k' gilt $L(G_{k'}) = L(G_k)$. Da M nach Voraussetzung auch dual monoton monoton arbeitet, müßte jetzt weiterhin

$$co\!-\!L(G_k) \cap co\!-\!L(G_k) \subseteq co\!-\!L(G_j) \cap co\!-\!L(G_k) \subseteq co\!-\!L(G_{k'}) \cap co\!-\!L(G_k)$$

gelten, im Widerspruch zu $co\!-\!L(G_k) \not\subseteq co\!-\!L(G_j)$. Damit ist das Corollar bewiesen.
q.e.d.

Wir weisen darauf hin, daß es bislang nicht gelungen ist, die Inklusion $KMON\!-\!TXT \subset KWMON\!-\!TXT$ zu beweisen oder zu widerlegen.

Der folgende Satz arbeitet eine wichtige Besonderheit des Lernens indizierter Familien heraus, die es gestattet, viele der nachfolgenden Beweise eleganter zu führen. Es zeigt sich nämlich, daß alle in Abschnitt 2.3 definierten Sprachlerntypen mit konsistenter Erkennung kombiniert werden können, ohne daß sich dadurch die prinzipielle Leistungsfähigkeit ändert.

Satz 4.18. *Für alle $A \in \{E, \lambda, K\}$ gilt:*

(1) $AWMON\!-\!CONS\!-\!TXT = AWMON\!-\!TXT$

(2) $AMON\!-\!CONS\!-\!TXT = AMON\!-\!TXT$

(3) $ASMON\!-\!CONS\!-\!TXT = ASMON\!-\!TXT$

(4) $AWMON^d\!-\!CONS\!-\!TXT = AWMON^d\!-\!TXT$

(5) $AMON^d\!-\!CONS\!-\!TXT = AMON^d\!-\!TXT$

(6) $ASMON^d\!-\!CONS\!-\!TXT = ASMON^d\!-\!TXT$

Beweis: Wir zeigen nur Aussage (1), da die anderen Aussagen analog bewiesen werden können. Offensichtlich gilt $AWMON\!-\!CONS\!-\!TXT \subseteq AWMON\!-\!TXT$ für alle $A \in$

$\{E, \lambda, K\}$. Zum Nachweis der verbleibenen Inklusion sei $\mathcal{L}$ eine beliebig fixierte indizierte Familie und M eine IIM mit $\mathcal{L} \subseteq AWMON{-}TXT(M)$ bzgl. eines Hypothesenraumes $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$. Wir definieren jetzt eine IIM $\hat{M}$ wie folgt.

Mögen $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ sein. Bei Eingabe von t_x simuliert $\hat{M}$ zunächst M bis zum Lesen von t_x. Hat M nach dem Lesen von t_x noch gar keine Hypothese produziert, so gibt $\hat{M}$ keine Hypothese aus und liest das nächste Wort auf dem Eingabeband. Andernfalls möge j_z die letzte von M bis zum Lesen von t_x ausgegebene Hypothese sein. Die Maschine $\hat{M}$ testet nun, ob $t_x^+ \subseteq L(G_{j_z})$. Wenn ja, so wird j_z ausgegeben. Gilt $t_x^+ \not\subseteq L(G_{j_z})$, so gibt $\hat{M}$ keine Hypothese aus und liest das nächste Wort auf dem Eingabeband.

Nach Konstruktion arbeitet $\hat{M}$ offenbar konsistent. Da M auf t gegen eine für L korrekte Hypothese j zu konvergieren hat, gibt $\hat{M}$ überdies auf t fast immer eine Hypothese aus und konvergiert ebenfalls gegen j.

q.e.d.

Wir wenden uns jetzt der Frage zu, wo die oberen bzw. unteren Grenzen der eben angegeben Hierarchien zu finden sind. Wie wie sehen werden, hängt die Beantwortung dieser Frage auch von den zugelassenen Hypothesenräumen ab. Der folgende Satz liefert ein besonders illustratives Beispiel dafür.

Satz 4.19. $SMON{-}TXT \setminus EWMON{-}TXT \neq \emptyset$

Beweis: Wir zeigen die Behauptung mit Hilfe der im Beweis von Satz 4.10, Aussage (3) angegebenen indizierten Familie $\mathcal{L}$. Es macht nun keine Mühe, sich zu überlegen, daß die dort im Beweis von Hilfssatz B definierte IIM insbesondere streng-monoton arbeitet.

q.e.d.

Die hier entwickelte Beweistechnik, die wir in dieser Form schon im Beweis von Satz 4.10 angewendet haben, entwickelt ihre volle Stärke beim Trennen von klassentreuer und klassenumfassender Inferenz. Ginge man hierbei mit klassischen Methoden vor, so hätte man explizit gegen alle induktiven Inferenzmaschinen und alle klassentreuen bzw. klassenumfassenden Hypothesenräume zu diagonalisieren, wodurch die resultierenden Beweise, gelinde gesagt, einen unübersichtlichen Charakter annähmen. Darüber hinaus erlauben die von uns vorgenommenen Reduktionen des Halteproblems auf die jeweils betrachteten Lernprobleme auch eine untere Abschätzung hinsichtlich des Grades der algorithmischen Unlösbarkeit der jeweiligen Lernprobleme. Diese Einsicht verdient aus mehreren Gründen Beachtung. Adleman und Blum (1991) haben gezeigt, daß der Grad der algorithmischen Unlösbarkeit von $\mathcal{R} \in EX$ *echt* unterhalb des Unlösbarkeitsgrades vom Halteproblem liegt. Während nun einerseits ein algorithmisch lösbares Lernpro-

blem vorliegt (z.B. $\mathcal{L} \in SMON-TXT$), wird durch die Hinzunahme einer zunächst natürlich erscheinenden Forderung (z.B. exaktes Lernen) das entstehende Lernproblem (z.B. $\mathcal{L} \in EWMON-TXT$) jeweils algorithmisch unlösbar. Dabei ist der Grad der algorithmischen Unlösbarkeit uniform größer oder gleich dem des Halteproblems und damit echt größer als der des Problems, alle rekursiven Funktionen im Limes zu lernen. Diese Resultate zeigen also insbesondere, daß die hier betrachteten Forderungen gravierende Auswirkungen auf die prinzipielle Lösbarkeit der studierten Lernprobleme haben können. Es sei noch angemerkt, daß es unserer Kenntnis nach nur eine Arbeit gibt, in der ein analoges Resultat im Bereich des Lernens rekursiver Funktionen gezeigt wurde, i.e. Freivalds, Kinber und Wiehagen (1992) (cf., Theorem 32).

Ein weiterer Vorteil unserer Technik offenbart sich bei einem näherem Blick auf die verwendeten indizierten Familien. Es zeigt sich nämlich, daß zunächst völlig kanonisch erscheinende Beschreibungen und Aufzählungen von uniform rekursiven Sprachen untauglich sein können, wenn sie im Bereich des algorithmischen Lernens eingesetzt werden. Es kommt vielmehr darauf an, eine dem jeweiligen Lernproblem adäquate Darstellung der zu lernenden Objekte zu entwerfen. Wir werden diesen Aspekt im Abschnitt 4.4 noch gründlicher beleuchten. Doch zunächst separieren wir streng-monotones bzw. dual streng-monotones klassenumfassendes Lernen von streng-monotoner bzw. dual streng-monotoner Inferenz.

Satz 4.20.

(1) $KSMON-TXT \setminus SMON-TXT \neq \emptyset$

(2) $KSMON^d-TXT \setminus SMON^d-TXT \neq \emptyset$

Beweis: Die gewünschte indizierte Familie definieren wir folgendermaßen. Für alle $k \in \mathbb{N}$ setzen wir:

$L_{2k} = \{a^{k+1}b^{2n} \mid n \in \mathbb{N}^+\}$

$L_{2k+1} = \{a^{k+1}b^{2n} \mid 1 \leq n \leq z\} \cup \{a^{k+1}b^{2z+1}\}$, wobei $z = \mu n[\Phi_k(k) \leq n]$.

und definieren die indizierte Familie $\mathcal{L} =_{df} (L_j)_{j\in\mathbb{N}}$. Das Wortproblem ist uniform rekursiv für alle $w \in \{a,b\}^*$ und alle $L \in \mathcal{L}$, da das Prädikat „$\Phi_i(x) \leq y$“ für alle $i,x,y \in \mathbb{N}$ uniform entscheidbar ist. Nach Konstruktion gilt weiterhin $L_{2k} = L_{2k+1}$, sofern $\varphi_k(k)$ nicht definiert ist. Wir zeigen die Behauptung des Satzes durch eine Reihe von Hilfssätzen.

***Hilfssatz* A.** $\mathcal{L} \notin SMON{-}TXT$.

Angenommen doch, dann existieren eine IIM M und ein klassentreuer Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ derart, daß $\mathcal{L} \subseteq SMON{-}TXT(M)$ bzgl. $\mathcal{G}$. Wir zeigen wieder, daß man jedes Programm von M uniform in eine Entscheidungsprozedur $\mathcal{A}$ zur Lösung des Haltproblems transformieren kann.

Möge $k \in \mathbb{N}$ sein. $\mathcal{A}$ arbeitet bei Eingabe k wie folgt:

(A1) Generiere sukzessive inkrementell wachsende Anfangsstücke des kanonischen Textes t von L_{2k}, solange bis das erste Anfangsstück, sagen wir t_z, mit $j = M(t_z)$ und $t_z^+ \subseteq L(G_j)$ gefunden wird.

(A2) Teste, ob $\Phi_k(k) \leq z$. Wenn ja, so gib „$\varphi_k(k)$ definiert“ aus. Andernfalls gib „$\varphi_k(k)$ nicht definiert“ aus.

Da M die Sprache L_{2k} insbesondere auf ihrem kanonischen Text inferieren muß, terminiert die Abarbeitung von Anweisung (A1). Danach ist z bekannt, so daß Anweisung (A2) effektiv ausgeführt werden kann. Wir haben noch zu zeigen, daß der Algorithmus $\mathcal{A}$ korrekt arbeitet. Offensichtlich ist jede Ausgabe der Form „$\varphi_k(k)$ definiert“ korrekt. Möge jetzt also $\Phi_k(k) > z$ gelten. Wir haben zu zeigen, daß $\varphi_k(k)$ nicht definiert ist.

Angenommen, $\varphi_k(k)$ wäre definiert. Dann existerte ein $y \in \mathbb{N}$ mit $\Phi_k(k) = y$. Somit erhalten wir aus der Definition der Sprachen L_{2k} und L_{2k+1} sofort die Ungleichheit beider. Da unser Hypothesenraum klassentreu und die Hypothese j nach Konstruktion überdies konsistent ist, haben wir jetzt folgende Fälle zu unterscheiden.

Fall 1. $L(G_j) = L_{2k}$

Nach Konstruktion gilt $y > z$. Damit ist t_z auch ein Anfangsstück des kanonischen Textes von L_{2k+1}. Da andererseits $L_{2k} \not\subseteq L_{2k+1}$ kann M auf dem kanonischen Text von L_{2k+1} schon nicht mehr streng-monoton arbeiten. Also darf dieser Fall nicht vorkommen.

Fall 2. $L(G_j) = L_{2k+1}$

Nach Konstruktion ist t_z Anfangsstück des kanonischen Textes von L_{2k}. Aus der Definition der Sprachen L_{2k} und L_{2k+1} ergibt sich aus der Annahme $\Phi_k(k) = y$ sofort $a^{k+1}b^{2y+1} \in L_{2k+1} \setminus L_{2k}$. Folglich ist $L_{2k+1} \not\subseteq L_{2k}$. Damit kann M aber auf dem kanonischen Text t von L_{2k} nicht mehr streng-monoton arbeiten. Aus diesem Widerspruch erhalten wir die Korrektheit der Ausgabe „$\varphi_k(k)$ nicht definiert“.

q.e.d. (Hilfssatz A)

***Hilfssatz* B.** $\mathcal{L} \notin SMON^d{-}TXT$.

Wie der folgende Satz zeigt, gilt $SMON^d{-}TXT \subset SMON{-}TXT$. Damit ergibt sich Hilfssatz B direkt aus Hilfssatz A.

***Hilfssatz* C.** $\mathcal{L} \in KSMON{-}TXT$.

Zuerst definieren wir einen klassenumfassenden Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$. Seien $k \in \mathbb{N}$ und $0 \leq n \leq 2$. Wir setzen

$$L(G_{3k+n}) = \begin{cases} L_{2k}, & \text{falls } n = 0 \\ L_{2k+1}, & \text{falls } n = 1 \\ L_{2k} \cap L_{2k+1}, & \text{falls } n = 2 \end{cases}$$

Offenbar ist das Wortproblem uniform im Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ entscheidbar. Damit ist $\mathcal{G}$ ein zulässiger Hypothesenraum. Wir definieren jetzt eine IIM M, die $\mathcal{L}$ streng-monoton bzgl. $\mathcal{G}$ lernt. Dazu mögen $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ sein. Wir setzen:

$M(t_x) =$ „Ist $x = 0$, so bestimme zunächst den relevanten Index k. Für $x \geq 0$ teste, ob $t_x^+ \subseteq L(G_{3k+2})$. Wenn ja, so gib $3k+2$ aus. Andernfalls arbeite (A) ab.

(A) Wenn $t_x^+ \subseteq L(G_{3k})$, so gib $3k$ aus.
Andernfalls gib $3k+1$ aus und lies das nächste Eingabewort."

Aus der Definition von M geht unmittelbar hervor, daß M höchstens einen Hypothesenwechsel ausführt. Dieser Wechsel erfolgt von der Hypothese $3k+2$ entweder zur Hypothese $3k$ oder zu $3k+1$. In beiden Fällen gilt $L(G_{3k+2}) \subseteq L(G_j)$, wobei entweder $j = 3k$ oder $j = 3k+1$. Somit arbeitet M streng-monoton und konvergiert gegen eine für die zu lernende Sprache korrekte Hypothese.

q.e.d. (Hilfssatz C)

***Hilfssatz* D.** $\mathcal{L} \in KSMON^d{-}TXT$.

Wir definieren einen klassenumfassenden Hypotheseraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ durch: Für alle $k \in \mathbb{N}$ und n mit $0 \leq n \leq 2$ sei

$$L(G_{3k+n}) = \begin{cases} L_{2k}, & \text{falls } n = 0 \\ L_{2k+1}, & \text{falls } n = 1 \\ L_{2k} \cup L_{2k+1}, & \text{falls } n = 2 \end{cases}$$

Wir definieren jetzt eine IIM M, die $\mathcal{L}$ dual streng-monoton bzgl. $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ lernt. Dazu mögen $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ sein. Informal gibt M auf t solange die Vereinigung

$L_{2k} \cup L_{2k+1}$ aus, bis im Text t ein Wort w derart gefunden wird, daß entweder $w \notin L_{2k}$ oder $w \notin L_{2k+1}$. Dann wird der inkonsistente Anteil aus der Vereinigung gestrichen und die neue Hypothese in jedem nachfolgenden Schritt ausgegeben. Formal setzen wir:

$M(t_x) =$ „Ist $x = 0$, so bestimme zunächst den relevanten Index k. Für $x \geq 0$ führe (A) aus.

(A) Teste, ob $t_x^+ \subseteq L(G_{3k+1})$. Ist dies nicht der Fall, so gib $3k$ aus und lies die nächste Eingabe. Andernfalls arbeite (B) ab.

(B) Wenn $t_x^+ \subseteq L(G_{3k})$, so gib $3k+2$ aus und lies die nächste Eingabe.
Sonst gib $3k+1$ aus und lies die nächste Eingabe.

Es macht jetzt keine Mühe $\mathcal{L} \subseteq KSMON^d\text{–}TXT(M)$ bzgl. $\mathcal{G}$ nachzuweisen.

q.e.d.

Die letzten Sätze gestatten jetzt unmittelbar die gewünschte Abschätzung der Leistungsfähigkeit streng-monotoner bzw. dual streng-monotoner Inferenz nach unten. Dabei liefern uns die folgenden Resultate gleichzeitig eine Antwort auf die Frage, ob es günstiger ist, ausschließlich durch Generalisieren oder durch Spezialisieren zu lernen. Interessanterweise ändert sich die Antwort in Abhängigkeit von der gewählten Formalisierung der Begriffe „ generalisieren“ und „spezialisieren“. Werden beide in ihrer strengsten Form interpretiert, so ist Lernen durch Generalisieren deutlich leistungsfähiger als Lernen durch Spezialisieren.

Satz 4.21.

(1) $FIN\text{–}TXT \subset ESMON\text{–}TXT \subset SMON\text{–}TXT \subset KSMON\text{–}TXT$

(2) $FIN\text{–}TXT = ESMON^d\text{–}TXT = SMON^d\text{–}TXT \subset KSMON^d\text{–}TXT$

(3) $KSMON^d\text{–}TXT \subset EWMON\text{–}TXT$

Beweis: Aus den Definitionen 13 und 17 ergibt sich $FIN\text{–}TXT \subseteq ESMON\text{–}TXT \subseteq SMON\text{–}TXT \subseteq KSMON\text{–}TXT$ unmittelbar. Nach Satz 4.20 ist die letzte Inklusion echt. Da sich $SMON\text{–}TXT \setminus ESMON\text{–}TXT \neq \emptyset$ direkt aus Satz 4.19 ergibt, bleibt nur noch $ESMON\text{–}TXT \setminus FIN\text{–}TXT \neq \emptyset$ zu zeigen. Dazu verwenden wir die folgende indizierte Familie $\mathcal{L}_{finvar}$. Sei L_{cs} eine beliebig fixierte unendliche kontext-sensitive Sprache über dem Alphabet $\{a, b\}$ und $\mathcal{L}_{fin}$ die kanonische Aufzählung aller endlichen Sprachen über $\{a, b\}$. Mit $\mathcal{L}_{finvar}$ bezeichnen wir dann die Familie aller Sprachen $L_{cs} \cup L_{fin}$, wobei

$L_{fin} \in \mathcal{L}_{fin}$, in kanonischer Aufzählung. Damit ist $\mathcal{L}_{finvar}$ offenbar eine indizierte Familie. Mit Hilfe der Charakterisierung finit erlernbarer indizierter Familien durch rekursive Tell-tale Mengen (cf., Satz 4.39) beweist man mühelos $\mathcal{L}_{finvar} \notin FIN{-}TXT$.

Wir definieren jetzt eine IIM M derart, daß $\mathcal{L}_{finvar} \subseteq ESMON{-}TXT(M)$. Mögen dazu $L \in \mathcal{L}_{finvar}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ sein. Wir setzen:

$M(t_x) =$ „Berechne $L^x_{fin} = \{w \mid w \in t^+_x \setminus L_{cs}\}$. Gib den kanonischen Index von $L^x_{fin} \cup L_{cs}$ aus.“

Nach Konstruktion arbeitet M offensichsichtlich streng-monoton. Aus der Definition der Familie $\mathcal{L}_{finvar}$ erhalten wir überdies direkt, daß M auf t gegen eine für L korrekte Nummer konvergiert. Damit ist (1) vollständig gezeigt.

Wir beginnen den Beweis von (2) mit dem Nachweis von $FIN{-}TXT = SMON^d{-}TXT$. Da $FIN{-}TXT \subseteq SMON^d{-}TXT$ offensichtlich ist, möge jetzt $\mathcal{L} \subseteq SMON^d{-}TXT(M)$ bzgl. eines klassentreuen Hypothesenraumes $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ gelten. Seien $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ derart, daß M nach dem Lesen von t_x zum ersten Mal eine Hypothese, sagen wir j_x, produziert. Angenommen, $L \neq L(G_{j_x})$. Aus der Definition der strengen Monotonie folgt sofort, daß dann $L \subset L(G_{j_x})$ gelten muß. Da $\mathcal{G}$ ein klassentreuer Hypothesenraum ist, haben wir zusätzlich $L(G_{j_x}) \in \mathcal{L}$. Nun muß M weiterhin L bei Belehrung durch t lernen. Somit existiert ein $r \in \mathbb{N}$, so daß $j_{x+r} = M(t_{x+r})$ und $L = L(G_{j_{x+r}})$. Wegen $L \subseteq L(G_{j_x})$ ist aber t_{x+r} auch Anfangsstück eines Textes für $L(G_{j_x})$, woraus sofort folgt, daß M auf keinem Text $\tilde{t}$ für $L(G_{j_x})$ mit Präfix t_{x+r} dual streng-monoton arbeiten kann. Deshalb muß die Annahme $L \neq L(G_{j_x})$ falsch sein, d.h. die erste von M produzierte Hypothese ist korrekt. Damit kann M mühelos in eine IIM $\hat{M}$ mit $\mathcal{L} \subseteq FIN{-}TXT(\hat{M})$ transformiert werden.

Der Rest von Aussage (2) folgt unmittelbar aus Satz 4.20.

Wir beweisen jetzt Aussage (3). Seien $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$, $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ und eine IIM M mit $\mathcal{L} \subseteq KSMON^d{-}TXT(M)$ bzgl. $\mathcal{G}$ beliebig fixiert. Dann gilt für alle $k, j \in \mathbb{N}$ entweder $L_k = L_j$ oder $L_k \# L_j$. Um dies zu sehen, nehmen wir das Gegenteil an. Dann gibt es $k, j \in \mathbb{N}$ mit $L_k \neq L_j$ und $L_k \setminus L_j = \emptyset$ oder $L_j \setminus L_k = \emptyset$. Also gilt entweder $L_k \subset L_j$ oder $L_j \subset L_k$. Möge o.B.d.A. $L_k \subset L_j$ sein. Dann kann L_j nicht dual streng-monoton von M gelernt werden. Das sehen wir folgendermaßen. Sei $t \in \mathit{Text}(L_k)$. Somit muß ein $x \in \mathbb{N}$ derart existieren, daß $j_x = M(t_{x+r})$ für alle $r \in \mathbb{N}$ und $L_k = L(G_{j_x})$. Da aber $t^+_x \subseteq L_j$ kann M auf keinem Text $\tilde{t} \in \mathit{Text}(L_j)$ mit $t_x \sqsubset \tilde{t}$ dual streng-monoton arbeiten.

Wir definieren schließlich eine IIM M, die $\mathcal{L}$ bzgl. $\mathcal{L}$ schwach-monoton lernt. Seien $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$. Wir setzen:

$M(t_x) =$ „Suche den kleinsten Index $j \leq x$ mit $t_x^+ \subseteq L_j$. Gib j aus. Gilt $t_x^+ \not\subseteq L_j$ für alle $j \leq x$, so gib keine Hypothese aus und lies die nächste Eingabe."

Mit den obigen Beobachtungen macht es nun keine Mühe, $\mathcal{L} \subseteq EWMON{-}TXT(M)$ zu zeigen.

q.e.d.

An dieser Stelle ist es an der Zeit, das Verhältnis von konservativer und schwach-monotoner Inferenz aufzuklären.

Satz 4.22. *Für alle $A \in \{E, \lambda, K\}$ gilt:*

$$AWMON{-}TXT = ACONSERVATIVE{-}TXT$$

Beweis: Die Inklusion $ACONSERVATIVE{-}TXT \subseteq AWMON{-}TXT$ ist klar, da eine konservative IIM nur gerechtfertigte Hypothesenwechsel ausführt.

Möge jetzt $\mathcal{L} \subseteq AWMON{-}TXT(M)$ bzgl. eines Hypothesenraumes $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ sein. Wir haben eine IIM $\hat{M}$ derart zu konstruieren, daß $\mathcal{L} \subseteq ACONSERVATIVE{-}TXT(\hat{M})$. $\hat{M}$ erhält man aus M, indem in jedem Schritt ein zusätzlicher Konsistenztest durchgeführt wird. Initial gibt $\hat{M}$ die erste von M bei Belehrung durch t produzierte Hypothese aus. Möge j die letzte von $\hat{M}$ nach dem Lesen von t_{x-1} ausgegebene Hypothese sein. Auf t_x testet $\hat{M}$ zunächst, ob $t_x^+ \subseteq L(G_j)$. Wenn ja, so wird j ausgegeben. Andernfalls gibt $\hat{M}$ die letzte von M auf t_x produzierte Hypothese aus.

Damit arbeitet $\hat{M}$ konservativ und konvergiert gegen die erste korrekte Hypothese von M.

q.e.d.

Wir wenden uns nun der Frage zu, wo die prinzipiellen Grenzen schwach-monotonen bzw. dual schwach-monotonen Lernens liegen. Die Antwort wird durch die folgenden Sätze geliefert. Insbesondere weisen wir, anknüpfend an die obigen Bemerkungen, darauf hin, daß sich das Verhältnis der Leistungsfähigkeit von Lernen durch Generalisieren und Lernen durch Spezialisieren umkehrt, wenn beide Begriffe in ihrer schwächsten Form interpretiert werden.

Satz 4.23.

(1) $EWMON{-}TXT \subset WMON{-}TXT \subset KWMON{-}TXT \subset LIM{-}TXT$

(2) $EWMON^d{-}TXT \subset WMON^d{-}TXT \subset KWMON^d{-}TXT = LIM{-}TXT$

Beweis: Satz 4.22 und Corollar 4.11 ergeben $EWMON{-}TXT \subset WMON{-}TXT$. Aus beweistechnischen Gründen zeigen wir als nächstes $KWMON{-}TXT \subset LIM{-}TXT$.

Nun gilt offenbar $KWMON-TXT \subseteq LIM-TXT$, so daß wir lediglich $LIM-TXT \setminus KWMON-TXT \neq \emptyset$ zu beweisen haben. Unter Verwendung von Satz 4.22 genügt somit der Nachweis von $LIM-TXT \setminus KCONSERVATIVE\text{–}TXT \neq \emptyset$. Wir definieren die separierende indizierte Familie $\mathcal{L}_{kconsv}$ wie folgt:

Für alle $k \in \mathbb{N}$ sei $L_{c(k,0)} = \{a^k b^n \mid n \in \mathbb{N}^+\}$ und für $j \geq 1$ unterscheiden wir folgende Fälle.

Fall 1. $\neg\Phi_k(k) \leq j$

Dann sei $L_{c(k,j)} = L_{c(k,0)}$.

Fall 2. $\Phi_k(k) \leq j$

Wir setzen

$$L_{c(k,j)} = \begin{cases} \{a^k b^m \mid 1 \leq m \leq 2\Phi_k(k) - j\}, & \text{falls } \Phi_k(k) \leq j < 2\Phi_k(k), \\ \{a^k b\}, & \text{sonst.} \end{cases}$$

Schließlich sei $\mathcal{L}_{kconsv} = (L_{c(k,j)})_{k,j \in \mathbb{N}}$. Zum Nachweis von $\mathcal{L}_{kconsv} \notin KCONSERVATIVE\text{–}TXT$ sind die im Beweis von Satz 4.10 entwickelten Ideen mutatis mutandis ausreichend. Wir skizzieren deshalb deshalb nur die einzelnen Schritte. Angenommen, es gibt einen Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ und eine IIM M derart, daß $\mathcal{L}_{kconsv} \subseteq KCONSERVATIVE\text{–}TXT(M)$. Dann kann mit Hilfe von M wieder das Halteproblem effektiv gelöst werden. Zur Entscheidung, ob $\varphi_k(k)$ definiert ist, generiert man den kanonischen Text t von $L_{c(k,0)}$, bis das kleinste z mit $t_z^+ \subseteq L(G_{M(t_z)})$ und $a^k b^{z+1} \in L(G_{M(t_z)})$ gefunden wurde. Anschließend testet man, ob $\Phi_k(k) \leq z$. Wenn ja, so ist $\varphi_k(k)$ definiert; andernfalls ist $\varphi_k(k)$ divergent.

Es bleibt zu zeigen, daß $\varphi_k(k)$ tatsächlich nicht definiert ist, falls $\neg\Phi_k(k) \leq z$. Angenommen, $\varphi_k(k)$ wäre definiert. Dann gäbe es ein $y > z$ mit $\Phi_k(k) = y$. Nach der Definition von $\mathcal{L}_{kconsv}$ existiert in diesem Fall aber eine Sprache $L \in \mathcal{L}_{kconsv}$ mit $L = t_z^+$. Damit ist t_z aber auch Anfangsstück eines Textes für L. Da jedoch $a^k b^{z+1} \in L(G_{M(t_z)}) \setminus L$, hat M auf diesem Anfangssstück eine für L übergeneralisierte Hypothese produziert. Aus diesem Widerspruch folgt die Behauptung unmittelbar.

Der Nachweis von $\mathcal{L}_{kconsv} \in LIM-TXT$ ist trivial, so daß wir hier auf ihn verzichten. Wir zeigen weiter unten das wesentlich schärfere Resultat $\mathcal{L}_{kconsv} \in EMON^d-TXT$.

Den Beweis von (1) beenden wir mit der Separation von $WMON-TXT$ und $KWMON-TXT$. Da die Inklusion $WMON-TXT \subseteq KWMON-TXT$ offensichtlich ist, genügt es, eine indizierte Familie $\mathcal{L}$ mit $\mathcal{L} \in KWMON-TXT \setminus WMON-TXT$ anzugeben. Nach Satz 4.22 ist die letzte Aussage äquivalent zu $\mathcal{L} \in KWMON-TXT \setminus$

CONSERVATIVE–TXT. Wir definieren die gewünschte Familie $\mathcal{L}$ wie folgt:

Für alle $k \in \mathbb{N}$ setzen wir $L_{c(k,0)} = \{a^k b^n \mid n \in \mathbb{N}^+\}$ und für $j \geq 1$ unterscheiden wir die folgenden Fälle:

Fall 1. $\neg\Phi_k(k) \leq j$

Dann setzen wir $L_{c(k,j)} = L_{c(k,0)}$.

Fall 2. $\Phi_k(k) \leq j$

Dann setzen wir

$$L_{c(k,j)} = \begin{cases} \{a^k b^m \mid 1 \leq m \leq 2\Phi_k(k) - j\} \cup \{a^k b^{z+2}\}, & \text{falls } \Phi_k(k) \leq j < 2\Phi_k(k), \\ \{a^k b\} \cup \{a^k b^{z+2}\}, & \text{sonst.} \end{cases}$$

Schließlich sei $\mathcal{L} = (L_{c(k,j)})_{k,j \in \mathbb{N}}$. Völlig analog wie oben beweist man jetzt $\mathcal{L} \notin$ *CONSERVATIVE–TXT*. Wir merken an, daß hierzu die Wörter der Form $a^k b^{z+2}$ nicht benötigt werden. Deren Bedeutung arbeiten wir im folgenden Nachweis von $\mathcal{L} \in KWMON-TXT$ heraus. Dazu definieren wir als erstes einen klassenumfassenden Hypothesenraum $\mathcal{G} = (G_{c(k,j)})_{k,j \in \mathbb{N}}$.

Für alle $k \in \mathbb{N}$ sei $G_{c(k,0)}$ eine kanonische Grammatik für $L_{c(k,0)}$. Für $j = 1$ definieren wir $G_{c(k,1)}$ durch die Bedingung $a^k b^z \in L(G_{c(k,1)})$ gdw. $\neg\Phi_k(k) \leq z - 1$. Schließlich sei $G_{c(k,j)}$ für alle $j \geq 2$ eine kanonische Grammatik für $L_{c(k,j-1)}$.

Die indizierte Familie $\mathcal{L}$ kann nun konservativ bzgl. $\mathcal{G}$ gelernt werden, indem auf jedem Text einer Sprache $L \in \mathcal{L}$ solange $c(k,1)$, für das relevante k, ausgegeben wird, bis eventuell $t_x^+ \not\subseteq L(G_{c(k,1)})$ verifiziert wird. Tritt dieser Fall ein, so ist $\varphi_k(k)$ definiert. Also kann y mit $\Phi_k(k) = y$ berechnet und anschließend die bzgl. Inklusion kleinste konsistente Hypothese effektiv ermittelt und ausgegeben werden. Damit ist (1) gezeigt.

Wir beginnen den Beweis von (2) mit folgendem Lemma.

***Lemma* A**. *Möge $\mathcal{L}$ eine indizierte Familie mit $\mathcal{L} \in LIM-TXT$ sein. Dann kann aus $\mathcal{L}$ effektiv ein Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ derart konstruiert werden, daß $\mathcal{L} \in WMON^d-TXT$ bzgl. $\mathcal{G}$.*

Möge $\mathcal{L} \subseteq LIM-TXT(\hat{M})$ bzgl. $\mathcal{L}$ für eine IIM $\hat{M}$ sein (cf., Satz 4.1). O.B.d.A. können wir voraussetzen, daß $\hat{M}$ auf jedem Text nach Ausgabe der ersten Hypothese in jedem Schritt eine Hypothese ausgibt. Der gewünschte Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ ist die kanonische Aufzählung aller endlichen Durchschnitte von Sprachen aus $\mathcal{L}$. Damit ist das Wortproblem für alle Sprachen $L(G_j)$ uniform entscheidbar (cf., Bucher und Maurer, 1984) und es gilt $Val\ \mathcal{L} \subseteq \{L(G_j) \mid j \in \mathbb{N}\}$. Wir definieren jetzt eine IIM M, die $\mathcal{L}$ dual

schwach-monoton bzgl. $\mathcal{G}$ inferiert. Seien $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$. Wir setzen:

$M(t_x) =$ „Simuliere $\hat{M}$ bis zum Lesen von t_x. Gibt $\hat{M}$ bis zum Lesen von t_{x-1} keine, jedoch auf t_x eine Hypothese, sagen wir j aus, so gib eine kanonische Nummer für L_j im Hypothesenraum $\mathcal{G}$ aus.

Andernfalls möge k die letzte von M bis zum Lesen von t_{x-1} produzierte Hypothese sein. Teste, ob $t_x^+ \subseteq L(G_k)$. Ist dies nicht der Fall, so gib die letzte von $\hat{M}$ bis zum Lesen von t_x produzierte Hypothese aus. Wenn der Konsistenztest erfüllt ist, verfahre wie folgt:

Teste, ob $\hat{M}(t_x) \neq \hat{M}(t_{x-1})$. Führt $\hat{M}$ keinen Hypothesenwechsel aus, so gib $M(t_{x-1})$ aus. Andernfalls gib eine kanonische Grammatik für $L(G_k) \cap L_{j_x}$ aus, wobei $j_x = \hat{M}(t_x)$."

Wir haben noch $\mathcal{L} \subseteq KWMON{-}TXT(M)$ zu zeigen.

Beh. 1. M arbeitet dual schwach-monoton.

Möge M einen Hypothesenwechsel ausführen. Ist dieser Wechsel durch Inkonsistenz bedingt, so haben wir nichts zu zeigen. Nach Konstruktion gilt für Hypothesenwechsel von k nach j mit $t_x^+ \subseteq L(G_k)$, daß j eine kanonische Nummer von $L(G_k) \cap L_{j_x}$ für $j_x = \hat{M}(t_x)$ ist. Also gilt $co{-}L(G_k) \subseteq co{-}L(G_j)$.

Beh. 2. Wenn M auf t konvergiert, sagen wir gegen z, so ist $L = L(G_z)$.

Angenommen, M konvergiert und es gilt $L \neq L(G_z)$. Dann muß nach Konstruktion von M offenbar $L \subset L(G_z)$ gelten, da M sonst eine Inkonsistenz entdeckte und folglich einen Hypothesenwechsel ausführte. Da M konvergiert hat, ergibt sich aus der Definition von M weiterhin, daß auch $\hat{M}$ konvergiert haben muß. Sei k der Limes von $(\hat{M}(t_x))_{x \in \mathbb{N}}$. Dann läßt die Definition von M nur zwei Möglichkeiten zu. Entweder konvergiert M gegen eine kanonische Nummer von L_k oder gegen eine kanonische Nummer von $L(G_m) \cap L_k$, wobei m die letzte von z verschiedene Hypothese von M ist. Da $L = L_k$ erhalten wir in diesem Fall jedoch sofort $L(G_z) \subseteq L$, im Widerspruch zur Annahme.

Beh. 3. M konvergiert auf t.

Diese Behauptung ergibt sich unmittelbar aus der Definition von M.

q.e.d. (Lemma A)

Somit haben wir $KWMON^d{-}TXT = LIM{-}TXT$ gezeigt.

Wir setzen den Beweis von Aussage (2) mit folgendem Lemma fort. Dabei knüpfen wir gedanklich an Angluins (1980a) Beweis von $\mathit{ECONSERVATIVE}$–$TXT \subset LIM{-}TXT$

an, verfeinern jedoch die verwendete Technik.

***Lemma* B.** $EWMON^d{-}TXT \subset WMON^d{-}TXT$

Unser Ziel besteht in der Konstruktion einer indizierten Familie $\mathcal{L} = (L_{c(k,j)})_{k,j\in\mathbb{N}}$ mit $\mathcal{L} \in WMON^d{-}TXT \setminus EWMON^d{-}TXT$. Möge $(M_j)_{j\in\mathbb{N}}$ die Goldsche (1967) Aufzählung aller IIMs sein. Wir entwerfen die Teilfamilien $(L_{c(k,j)})_{j\in\mathbb{N}}$ derart, daß mindestens eine Sprache $L_{c(k,j)}$ nicht dual schwach-monoton von M_k bzgl. $\mathcal{L}$ gelernt wird.

Für alle $k \in \mathbb{N}$ und $j = 0$ setzen wir $L_{c(k,0)} = \{a^k b^n \mid n \in \mathbb{N}\}$. Ist $j \geq 1$ so wird $L_{c(k,j)}$ als Teilmenge von $\{a^k b^n \mid n \in \mathbb{N}\} \cup \{b^k a^n \mid n \in \mathbb{N}\}$ gemäß folgender Konstruktion definiert. Möge $t = a^k, a^k b, a^k b^2, ...$ der lexikographisch geordnete Text von $L_{c(k,0)}$ sein. Wir simulieren zunächst M_k bis zum Lesen von t_x für $x = 0, 1, ..., j$.

Fall 1. Es gibt kein $x \leq j$, so daß $M_k(t_x) = c(k,0)$.

Dann setzen wir $L_{c(k,j)} = \{b^k\}$.

Fall 2. Es gibt ein $x \leq j$ mit $M_k(t_x) = c(k,0)$.

Möge $\hat{x}$ das kleinste $x \leq j$ mit $M_k(t_x) = c(k,0)$ sein. Dann setzen wir $L' = \{a^k b^n \mid n \in \mathbb{N},\ n \leq \hat{x}+1\}$. Weiterhin sei $\hat{t} = a^k, a^k b, ..., a^k b^{\hat{x}+1}, a^k b^{\hat{x}+1}, ...$, d.h. der Text, der aus dem Anfangsstück $t_{\hat{x}+1}$ durch unendlich oftmaliges Wiederholen des letzten Wortes von $t_{\hat{x}+1}$ entsteht. Die folgende Berechnung wird durchgeführt, um zu bestimmen, welche Wörter der Form $b^k a^n$ zu $L_{c(k,j)}$ gehören sollen.

Wir definieren dazu $\hat{L}$ durch die Bedingung:
$b^k a^n \in \hat{L}$ gdw. es ein y mit $\hat{x} < y \leq n$ derart gibt, daß $M_k(\hat{t}_y) = c(k,u)$ für ein $u \geq j$.

Schließlich setzten wir $L_{c(k,j)} = L' \cup \hat{L}$. Offenbar ist $\mathcal{L} = (L_{c(k,j)})_{k,j\in\mathbb{N}}$ nach Konstruktion eine indizierte Familie.

Beh. 1. $\mathcal{L} \notin EWMON^d{-}TXT$

Angenommen doch; dann gibt es eine IIM M, die $\mathcal{L}$ dual schwach-monoton bzgl. $\mathcal{L}$ lernt. Diese IIM muß in der Aufzählung $(M_j)_{j\in\mathbb{N}}$ aller IIM vorkommen, sagen wir an der Stelle k. Also gilt $\mathcal{L} \subseteq EWMON^d{-}TXT(M_k)$. Wir betrachten zunächst das Verhalten von M_k auf dem lexikographisch geordneten Text t für $L_{c(k,0)}$. Wenn kein $x \in \mathbb{N}$ mit $M_k(t_x) = c(k,0)$ existiert, so wird $L_{c(k,0)}$ nicht bei Textbelehrung gelernt und wir sind fertig. Möge also $\hat{x}$ das kleinste x mit $M_k(t_x) = c(k,0)$ sein. Da $t^+_{\hat{x}+1} \subseteq L_{c(k,\hat{x})} \neq L_{c(k,0)}$ muß M_k jetzt auf dem oben definierten Text $\hat{t}$ für ein $y > \hat{x}$ eine Hypothese $c(k,u)$ mit $u > \hat{x}$ produzieren, da andernfalls $L_{c(k,\hat{x})}$ nicht gelernt wird. Nach Konstruktion kann dieser Hypothesenwechsel nicht durch Inkonsistenz mit den vorliegenden Daten verursacht werden. Also muß in diesem Falle $co{-}L_{c(k,0)} \subseteq co{-}L_{c(k,u)}$ gelten; im Widerspruch zu

$b^k a^n \in co\text{-}L_{c(k,0)} \setminus co\text{-}L_{c(k,u)}$ für alle $n \geq y$, wobei y der Bedingung $c(k,u) = M_k(\hat{t}_y)$ genügt.

Beh. 2. $\mathcal{L} \in WMON^d\text{-}TXT$

Wir definieren zunächst einen klassentreuen Hypothesenraum als indizierte Familie $\hat{\mathcal{L}} = (\hat{L}_j)_{j \in \mathbb{N}}$. Für alle $j \in \mathbb{N}$ sei $k = \lfloor j/3 \rfloor$. Wir setzen

$$\hat{L}_j = \begin{cases} L_{c(k,0)}, & \text{falls } j \equiv 0 \ mod\ 3, \\ \{b^k\}, & \text{falls } j \equiv 1 \ mod\ 3. \end{cases}$$

und für $j \equiv 2 \ mod\ 3$ definieren wir $\hat{L}_j$ durch die Bedingungen

(B1) $a^k b^n \in \hat{L}_j$ gdw. $n \leq \mu y[M_k(t_y) = c(k,0)] + 1$ und

(B2) $b^k a^n \in L_j$ gdw. $n \geq \mu y[M_k(t_y) = c(k,0)]+1$ und es gibt z mit $\mu y[M_k(t_y) = c(k,0)]+1 \leq z \leq n$, so daß $M_k(\hat{t}_z) = c(k,u)$ mit $u \geq \mu y[M_k(t_y) = c(k,0)] + 1$. Hierbei ist t der lexikographisch geordnete Text von $L_{c(k,0)}$ und $\hat{t}$ der dazu korrespondierende Text aus der obigen Konstruktion.

Wir merken an, daß wir jetzt für jedes k die drei möglichen Varianten für Sprachen in der Familie $\mathcal{L}$ auch in die Familie $\hat{\mathcal{L}}$ aufgenommen haben, jedoch in anderer Darstellung und Aufzählung.

Die gewünschte IIM $\hat{M}$ berechnet auf einem Text t einer Sprache $L \in \mathcal{L}$ zunächst das relevante k und gibt dann folgende Hypothesen aus. Ist $t_x^+ = \{b^k\}$, so wird $3k+1$ ausgegeben.
Sofern $t_x^+ \neq \{a^k\}$ ist, gibt $\hat{M}$ solange j mit $j = 3k+2$ als Hypothese aus, bis eine Inkonsistenz mit den eingelesenen Daten entdeckt wird.
Danach wird stets $3k$ ausgegeben.

Unter Verwendung der Tatsache, daß $\hat{L}_{3k} = \hat{L}_{3k+2}$, sofern es kein x mit $M_k(t_x) = c(k,0)$ gibt, überzeugt man sich leicht von $\mathcal{L} \subseteq WMON^d\text{-}TXT(\hat{M})$ bzgl. $\hat{\mathcal{L}}$.

q.e.d. (Lemma B)

Die Inklusion $WMON^d\text{–}TXT \subset KWMON^d\text{–}TXT$ wurde dem Autor von E.B. Kinber (1993) mitgeteilt, so daß wir hier auf die Präsentation des Beweises verzichten.

q.e.d.

Abschließend in diesem Abschnitt klären wir noch die Beziehung zwischen monotonem und dual monotonem Lernen. Damit vergleichen wir gleichzeitig die Leistungskraft von Algorithmen, die durch Generalisieren bzw. Spezialisieren lernen. Interessanterweise er-

gibt sich wieder ein neues Bild. Die Entscheidung, welcher Lernstrategie der Vorzug zu geben ist, hängt diesmal vom zu lösenden Lernproblem ab, wie der folgende Satz zeigt.

Satz 4.24.

(1) $EMON\text{-}TXT \# KMON^d\text{-}TXT$

(2) $EMON^d\text{-}TXT \# KMON\text{-}TXT$

Beweis: Zum Beweis dieses Satzes genügt es die folgenden beiden Fakten nachzuweisen

(α) $EMON^d\text{-}TXT \setminus KMON\text{-}TXT \neq \emptyset$ und

(β) $EMON\text{-}TXT \setminus KMON^d\text{-}TXT \neq \emptyset$.

Zum Nachweis von (α) genügt es nach Satz 4.23 eine IIM M derart anzugeben, daß $\mathcal{L}_{kconcv} \subseteq EMON^d\text{-}TXT(M)$. Mögen $L \in \mathcal{L}_{kconcv}$, $t \in Text(L)$ und $x \in \mathbb{N}$ sein. Nach Konstruktion von $\mathcal{L}_{kconcv}$ existiert somit ein eindeutig bestimmtes k, so daß $t_x^+ \subseteq L_{c(k,0)}$ für alle x. Wie man leicht sieht determiniert das erste Wort in t schon k. Wir definieren:

$M(t_x) =$ „Teste für alle $j = 1, ..., x$, ob $a^k b^x \notin L_{c(k,j)}$. Wenn der Test für mindestens ein j erfüllt ist, so arbeite (A) ab. Andernfalls beginne (B).

(A) Ist $card(t_x^+) = 1$, so berechne das kleinste y mit $a^k b^2 \notin L_{c(k,y)}$. Gib $c(k, y)$ aus. Andernfalls sei $z = max\{n \mid n \leq x,\ t_x^+ \subseteq L_{c(k,n)}\}$. Gib $c(k, z)$ aus.

(B) Gib $c(k, 0)$ aus."

Da $\mathcal{L}_{kconcv}$ eine indizierte Familie ist, können alle Anweisungen in der Definition von M effektiv ausgeführt werden. Somit ist M eine IIM. Wir haben noch zu zeigen, daß M die Sprache L dual monoton bei Belehrung durch Text t lernt. Dazu unterscheiden wir folgende Fälle.

Fall 1. $\varphi_k(k)$ divergiert.

Dann ist nach Konstruktion $L_{c(k,j)} = L_{c(k,0)}$ für alle $j \in \mathbb{N}$. Somit führt M stets (B) aus und produziert nur die korrekte Hypothese $c(k, 0)$.

Fall 2. $\varphi_k(k)$ konvergiert.

Möge z die kleinste Zahl n mit $\Phi_k(k) \leq n$ sein. Wir haben folgende Unterfälle zu betrachten.

Unterfall 2.1. $card(L) \leq z$

Nach Definition gibt M initial eventuell $c(k, 0)$ aus und arbeitet spätestens nach dem Lesen von t_z Anweisung (A) ab. Da $t \in Text(L)$ und L endlich muß überdies ein $r \in \mathbb{N}$

derart existieren, daß $t_r^+ = L$. Es ist nun leicht zu sehen, daß $M(t_r) = c(k, z+y)$ für ein $y \in \mathbb{N}$ und daß $L_{c(k,z+y)} = L$. Diese Hypothese $c(k, z+y)$ wird nach Konstruktion auch in jedem folgenden Lernschritt ausgegeben. Damit lernt M die Sprache L. Zum Nachweis der dualen Monotonie verifiziert man unmittelbar $L_{c(k,0)} \cup L \supseteq L_{c(k,z)} \cup L \supseteq ... \supseteq L_{c(k,z+y)} \cup L$, da $L_{c(k,0)} \supset L$ und $L_{c(k,z+y)} \subseteq L$ für alle $y \in \mathbb{N}$. Unter Verwendung der de Morganschen Regeln ergibt sich direkt die duale Monotonie.

Unterfall 2.2. $card(L) = \infty$

Dann muß $L = L_{c(k,0)}$ gelten. Offensichtlich gibt es somit ein $y > z$ derart, daß $t_y^+ \setminus L_{c(k,z)} \neq \emptyset$. Da $L_{c(k,z-1)} = L_{c(k,0)} = L$, ist $M(t_y) = c(k, z-1)$ und somit korrekt. Die duale Monotonie ergibt sich wie oben trivial, da diesmal $L = L_{c(k,0)} \supseteq L_{c(k,r)}$ für alle $r \in \mathbb{N}$. Damit ist (α) gezeigt.

Zum Beweis von (β) verwenden wir die folgende indizierte Familie. Sei $\Sigma = \{a, b, c\}$. Wir setzen $L_0 = \{a\}^*$. Für alle $k \geq 1$ sei $L_k = \{a^z \mid 1 \leq z \leq k\} \cup \{b^z \mid k < z\}$ und für $m > k \geq 1$ setzen wir $L_{k,m} = \{a^z \mid 1 \leq z \leq k\} \cup \{b^z \mid k < z \leq m\} \cup \{c^m\}$. Schließlich möge $\mathcal{L}$ die kanonische Aufzählung aller Sprachen L_k und $L_{k,m}$ sein. Man kann nun sogar schärfer $\mathcal{L} \in EMON{-}TXT \setminus KMON^d{-}INF$ zeigen (cf., Lange, Zeugmann und Kapur, 1992).

q.e.d.

Damit beenden wir zunächst unseren Exkurs über das Lernen mit Textbelehrung. Der nachfolgende Abschnitt behandelt die Frage, inwieweit sich die bislang erzielten Resultate auf den Fall des Lernens mit Informantbelehrung übertragen lassen.

4.2. Lernen bei Informantbelehrung

Wir untersuchen in diesem Abschnitt erneut die Frage, wie nichtmonotones Denken im Lernprozeß zu modellieren ist. Dabei interessiert uns vor allem, ob die oben erzielten Einsichten ihre Gültigkeit behalten, wenn der Lernprozeß bei Informantbelehrung durchgeführt wird. Um die Arbeit in ihren Grenzen nicht zu sprengen, haben wir uns entschlossen, hier nur den klassentreuen Fall zu behandeln. Für eine umfassende Darstellung monotonen Lernens bei Informantbelehrung verweisen wir auf Lange und Zeugmann (1993b). Wie unser erster Satz zeigt, gibt es nun sowohl Gemeinsamkeiten mit als auch Unterschiede zum Lernen bei Textbelehrung.

Satz 4.25.

(1) $SMON{-}INF \subset MON{-}INF \subset WMON{-}INF$

(2) $SMON^d\text{–}INF \subset MON^d\text{–}INF \subset WMON^d\text{–}INF$

(3) $MON\text{–}INF \ \# \ MON^d\text{–}INF$

(4) $SMON\text{–}INF \ \# \ SMON^d\text{–}INF$

Beweis: Aus Platzgründen führen wir den Beweis hier nicht und verweisen statt dessen auf Lange und Zeugmann (1993b).

Wir möchten noch darauf hinweisen, daß die im letzten Satz etablierten echten Inklusionen auch dann noch gelten, wenn man sich auf das Lernen indizierter Familien regulärer Sprachen beschränkt.

Der nächste Satz deckt weitere Unterschiede zum Lernen bei Textbelehrung auf. Insbesondere sehen wir, daß dual schwach-monotones Lernen deutlich an Leistungsfähigkeit gewinnt.

Satz 4.26.

(1) $FIN\text{–}INF \subset SMON\text{–}INF$

(2) $FIN\text{–}INF \subset SMON^d\text{–}INF \subset MON\text{–}INF \cap MON^d\text{–}INF$

(3) $WMON\text{–}INF = WMON^d\text{–}INF = LIM\text{–}INF$

(4) $CONSERVATIVE\text{–}INF = LIM\text{–}INF$

Beweis: Nach Definition sind die einfachen Inklusionen $FIN\text{–}INF \subseteq SMON\text{–}INF$ und $FIN\text{–}INF \subseteq SMON^d\text{–}INF$ klar. Da wir nach Satz 4.25 weiterhin wissen, daß $SMON\text{–}INF \# SMON^d\text{–}INF$ ist, können wir sofort $FIN\text{–}INF \neq SMON\text{–}INF$ sowie $FIN\text{–}INF \neq SMON^d\text{–}INF$ schlußfolgern. Damit erhalten wir (1) und den ersten Teil von (2). Für den zweiten Teil von Aussage (2) ist lediglich $SMON^d\text{–}INF \subseteq MON\text{–}INF$ zu zeigen, da $SMON^d\text{–}INF \subseteq MON^d\text{–}INF$ klar ist und die Echtheit der Inklusion sofort aus Aussage (4) von Satz 4.25 folgt. Möge also $\mathcal{L} \in SMON^d\text{–}INF$ bzgl. $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ sein. Dann existiert eine IIM M, die $\mathcal{L}$ dual streng-monoton lernt. Somit gilt für alle $L \in \mathcal{L}$, alle $i \in Info(L)$ und alle $x \in \mathbb{N}$: Wenn $j_x = M(i_x)$, so ist $L \subseteq L(G_{j_x})$. Also gilt für alle überhaupt auf i von M produzierten Hypothese j_x stets $L \cap L(G_{j_x}) = L$, woraus unmittelbar folgt, daß M auch monoton arbeitet.

Zum Beweis von (3) und (4) genügt es, sich zu überlegen, daß jede indizierte Familie $\mathcal{L}$ bei Informantbelehrung bzgl. des Hypothesenraumes $\mathcal{L}$ durch das Goldsche (1967) Verfahren „Identifizierung durch Aufzählung“ gelernt werden kann. Dabei werden nur durch

Inkonsistenz bedingte Hypothesenwechsel durchgeführt. Aus dieser Beobachtung folgen (3) und (4) unmittelbar.

q.e.d.

Unser bisheriger kurzer Exkurs durch die Welt des Lernes indizierter Familien bei Informantbelehrung hat bereits eine Reihe von Resultaten geliefert, die wir in folgendem Bild zusammenfassen.

Monotones Lernen bei Informantbelehrung

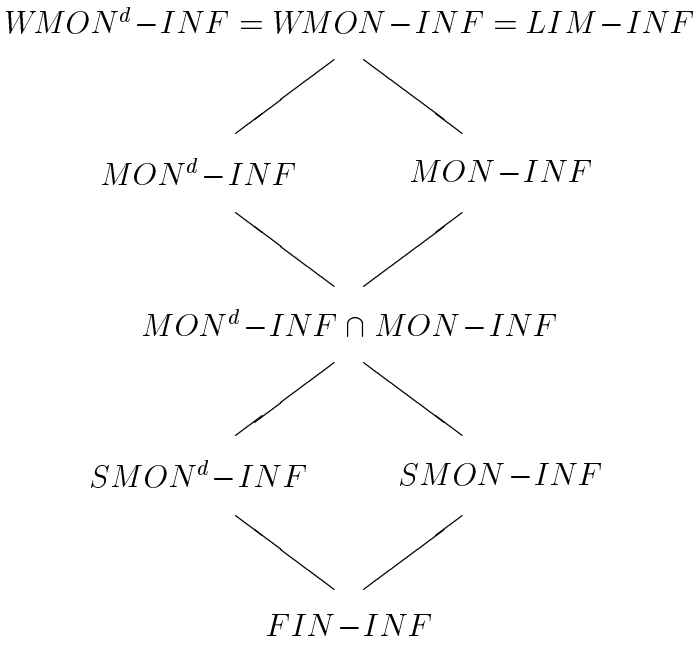

Abbildung 9

Abschließend vergleichen wir die Leistungsfähigkeit von monoton bzw. dual monoton arbeitenden Lernalgorithmen. Gold (1967) konnte bereits zeigen, daß Lernen aus positiven Daten allein wesentlich schwächer ist als Lernen bei Informantbelehrung. Somit ist es nicht verwunderlich, alle Typen monotonen bzw. dual monotonen Lernens bei Textbelehrung als leistungsschwächer zu charakterisieren als die entsprechenden Inferenzmodi bei Informantbelehrung. Interessanter hingegen ist die Frage, ob sich die Realisierung von stärkeren Monotonieforderungen durch den Übergang von Text- zu Informantbelehrung erkaufen läßt. Satz 4.16 deutet bereits darauf hin, daß dies im allgemeinen nicht der Fall sein wird. Wie unser nächster Satz zeigt, ist dieser Eindruck richtig.

Satz 4.27.

(1) $MON{-}INF \quad \# \quad WMON{-}TXT$

(2) $MON{-}INF \quad \# \quad LIM{-}TXT$

(3) $SMON{-}INF \quad \# \quad MON{-}TXT$

(4) $FIN{-}INF \quad \# \quad SMON{-}TXT$

Beweis: Nach Satz 4.16 wissen wir schon, daß $EWMON{-}TXT \setminus KMON{-}INF \neq \emptyset$. Damit genügt für den Nachweis von (1) und (2) die Angabe einer indizierten Familie $\mathcal{L}$ mit $\mathcal{L} \in MON{-}INF \setminus LIM{-}TXT$. Wie man leicht nachrechnet leistet gerade die Familie $\mathcal{L}$ aus dem Beweis von Satz 4.3, Hilfssatz A mit der dort angegebenen IIM das Verlangte.

Zum Nachweis von (3) und (4) nutzen wir zunächst die Klasse $\mathcal{L}_{dmon}$ aus dem Beweis von Satz 4.3. Da $\mathcal{L}_{dmon} \in FIN{-}INF \setminus MON{-}TXT$ gilt natürlich erst recht $\mathcal{L}_{dmon} \in FIN{-}INF \setminus SMON{-}TXT$ und $\mathcal{L}_{dmon} \in SMON{-}INF \setminus MON{-}TXT$. Nach Satz 4.16, Aussage (1) gilt auch $MON{-}TXT \setminus SMON{-}INF \neq \emptyset$, womit (3) und (4) gezeigt sind.

q.e.d.

Der letzte Satz hat folgende direkte Konsequenzen.

Corollar 4. 28.

(1) $SMON{-}TXT \subset SMON{-}INF$

(2) $MON{-}TXT \subset MON{-}INF$

(3) $WMON{-}TXT \subset WMON{-}INF$

Mit dem nachstehenden Satz knüpfen wir inhaltlich nochmal an die Bemerkungen vor Satz 4.27 an. Gleichzeitig beantworten wir damit die Frage, ob sich Aussage (2) dieses Satzes nicht verschärfen läßt. Das folgende Resultat beleuchtet weiterhin die Schärfe der Forderung, streng-monoton Lernen zu müssen, in besonders illustrativer Weise.

Satz 4.29. *Für alle $A \in \{E, \lambda, K\}$ gilt:*

$$ASMON{-}INF \quad \subset \quad AWMON{-}TXT$$

Beweis: Nach Satz 4.16 ist $AWMON{-}TXT \setminus ASMON{-}INF \neq \emptyset$ für alle $A \in \{E, \lambda, K\}$. Somit genügt der Nachweis von $ASMON{-}INF \subseteq AWMON{-}TXT$ für alle $A \in \{E, \lambda, K\}$. Da die Beweise für alle drei Fälle analog verlaufen, zeigen wir hier nur den

klassentreuen Fall. Mögen also $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ ein klassentreuer Hypothesenraum und $\hat{M}$ eine IIM mit $\mathcal{L} \subseteq SMON{-}INF(\hat{M})$ bzgl. $\mathcal{G}$ sein. Wir überlegen uns zuerst, daß $\hat{M}$ durch eine IIM M ersetzt werden kann, die $\mathcal{L}$ streng-monoton bei Informantbelehrung inferiert und die auf jeder endlichen Folge von Wörtern aus $\Sigma^* \times \{+, -\}$ definiert ist. Dies sieht man wie folgt.

Möge $i_x = (s_0, b_0), ..., (s_x, b_x)$ eine beliebig fixierte endliche Folge sein. Dann testet die IIM M zunächst, ob es eine Sprache $L_k \in \mathcal{L}$ mit $k \leq x$ und $i_x^+ \subseteq L_k$ sowie $i_x^- \subseteq co{-}L_k$ gibt. Wenn das nicht der Fall ist, so liest M das nächste Paar auf dem Eingabeband. Sollte dort kein weiteres Paar stehen, so arbeitet M in der Art weiter, als ob dort (s_x, b_x) stünde. Wenn dagegen eine Sprache L_k gefunden wird, so simuliert M auf i_x die Maschine $\hat{M}$. Wir weisen darauf hin, daß $\hat{M}$ auf i_x definiert sein muß, da andernfalls ein Anfangsstück eines Informanten für L_k existierte, auf dem $\hat{M}$ nicht korrekt arbeitete.

Wir haben noch zu zeigen, daß $\mathcal{L} \subseteq SMON{-}INF(M)$. Seien dazu $L \in \mathcal{L}$ und $i \in Info(L)$. Weiterhin sei $y = \mu z[L_z = L]$. Folglich findet M für alle $x \geq y$ nach dem Lesen von i_x mindestens eine Sprache L_k mit $k \leq x$ und $i_x^+ \subseteq L_k$ sowie $i_x^- \subseteq co{-}L_k$. Somit arbeitet M für alle $x \geq y$ genauso wie $\hat{M}$ und lernt deshalb L. Ist $x < y$, so verhält sich M entweder wie $\hat{M}$ oder M liest mehr und mehr Eingaben und gibt dabei keine Hypothese aus. Dabei kann es natürlich passieren, daß endlich viele der von $\hat{M}$ auf i produzierten Hypothesen von M übersprungen werden. Da Mengeninklusion eine transitive Relation ist, wird dadurch die strenge Monotonie der Hypothesenfolge nicht beeinträchtigt.

Wir haben jetzt eine IIM M' derart zu konstruieren, daß M' die Familie $\mathcal{L}$ bei Textbelehrung schwach-monoton inferiert. Mögen $L \in \mathcal{L}$, $t \in Text(L)$ und $x \in \mathbb{N}$ sein. Wir definieren M' wie folgt:

$M'(t_x) =$ „Arrangiere die in t_x enthaltenen Wörter in lexikographischer Ordnung ohne Wiederholungen. Möge $t'_k = w_0, ..., w_k$ die erhaltene Folge sein. Berechne
$A_x =_{df} \{(w_0, +), ..., (w_k, +)\} \cup \{(v, -) | v \in \Sigma^* \setminus t_x^+,\ |v| \leq max\{x, |w_k|\}\}$
Berechne $i_{z(x)}$, die lexikographische Aufzählung der Elemente von A_x bzgl. der ersten Komponente.
(* $i_{z(x)}$ ist nicht notwendigerweise ein Anfangsstück eines Informanten für L, da in $i_{z(x)}$ ein Paar $(v, -)$ mit $v \in L$ vorkommen kann. Wenn x jedoch groß genug ist, so gibt es stets ein $e \in \mathbb{N}$, so daß die Präfixe i_y von $i_{z(x)}$ für alle $y \leq e$ Anfangsstück eines Informanten für L sind. *)
Simuliere die IIM M bis entweder $i_{z(x)}$ vollständig gelesen wurde oder die erste Hypothese j_r, $r \leq z$ mit $i_{z(x)}^+ \subseteq L(G_{j_r})$ und $i_r^- \subseteq co{-}L(G_{j_r})$ gefunden wird.

(* Man beachte, daß für jedes Paar $(w,+) \in i^+_{z(x)}$ tatsächlich $w \in L$ gilt *)
Wenn das erste j_r gefunden wurde, so gib j_r aus. Andernfalls gib keine Hypothese aus und lies die nächste Eingabe."

Wir haben noch zu zeigen, daß M' das Verlangte leistet.

Beh. 1. M' konvergiert gegen eine Zahl j mit $L = L_j$.

Möge i^{ord} der lexikographisch geordnete Informant für L sein. Nach Voraussetzung lernt M die Sprache L auf i^{ord}. Somit muß ein kleinstes $m \in \mathbb{N}$ derart existieren, daß $j_m = M(i^{ord}_m)$ und $L = L(G_{j_m})$. Da $t \in \mathit{Text}(L)$, muß es ein x_0 geben, so daß $i^{ord,+}_m \subseteq t^+_x$ und $i^{ord,-}_m \subseteq i^-_{z(x)}$ für alle $x \geq x_0$, wobei $i_{z(x)}$ die aus t^+_x berechneten Anfangsstücke eventueller Informanten für L sind. Da $L = L(G_{j_m})$, erhalten wir:

$$i^+_{z(x)} \subseteq L(G_{j_m}) \text{ und } i^-_m \subseteq co{-}L(G_{j_m}) \text{ für alle } x \geq x_0.$$

Damit ist $j_m = M'(t_x)$ für alle $x \geq x_0$. Somit konvergiert M' gegen eine für L korrekte Hypothese.

Beh. 2. M' arbeitet schwach-monoton.

Nach Voraussetzung arbeitet M auf jedem Informanten für L streng-monoton. Wenn M also auf einem Anfangsstück i_n überhaupt eine Hypothese ausgibt, so muß in Übereinstimmung mit unserer Konstruktion folgendes gelten:

(1) Es gibt ein $k \leq n$, so daß $L_k \in \mathcal{L}$ und $i^+_n \subseteq L_k$ sowie $i^-_n \subseteq co{-}L_k$.

(2) Aus $j_n = M(i_n)$ folgt $L(G_{j_n}) \subseteq L_k$.

Da weiterhin stets $i^+_{z(x)} \subseteq L$ und co–L eingeschränkt auf Wörter der oben definierten Länge Teilmenge von i^-_n ist, sind alle betrachteten Anfangsstücke $i_{z(x)}$ entweder Anfangsstücke eines Informanten für L oder einer zur Familie $\mathcal{L}$ gehörenden Teilsprache L_k von L. Somit ist für jede überhaupt von M' ausgegebene Hpothese j die Beziehung $L(G_j) \subseteq L$ erfüllt. Es genügt daher für den weiteren Beweis zu zeigen, daß jeder Hypothesenwechsel von M' durch eine Inkonsistenz bzgl. des vorliegenden Textes verursacht wird. Möge j_r die von M' durch Simulation von M bis zum Lesen von i_r gefundene Hypothese mit $i^+_{z(x)} \subseteq L(G_{j_r})$ und $i^-_r \subseteq co{-}L(G_{j_r})$ sein. Wir unterscheiden zwei Fälle.

Fall 1. $i_r \sqsubseteq i^{ord}_m$

Wenn j_r nicht die Hypothese ist gegen die M' konvergiert, so kann ein Hypothesenwechsel nach Definition von M' nur dadurch verursacht werden, daß $i^+_{z(y)} \not\subseteq L(G_{j_r})$ für

ein $y > x$ verifiziert wird. Nach Konstruktion der Anfangsstücke $i_{z(x)}$ kann dieser Fall nur eintreten, wenn in t ein Wort w mit $w \notin L(G_{j_r})$ vorkommt. Also ist in diesem Fall der Hypothesenwechsel durch Inkonsistenz mit t erzwungen.

Fall 2. $i_r \not\sqsubseteq i_m^{ord}$

Nach Konstruktion kann dieser Fall nur eintreten, wenn in i_r ein Paar $(v, -)$ mit $v \in L$ vorkommt. Da $i_r^- \subseteq co-L(G_{j_r})$, gilt somit $v \notin L(G_{j_r})$. In diesem Fall wird also die Hypothese gewechselt, wenn ein Wort w im Text erscheint, so daß $(w, -) \in i_r^-$ enthalten ist oder ein Wort w mit $w \notin L(G_{j_r})$ in t auftritt. Wie auch immer, der Hypothesenwechsel ist durch Inkonsistenz bzgl. des vorliegenden Textes erzwungen.

q.e.d.

Betrachten wir jetzt anstelle streng-monotoner Inferenz dual streng-monotones Lernen, so ergibt sich ein völlig anderes Bild, das unter anderem auch zeigt, wie sensitiv dual streng-monotones Lernen gegenüber dem im Lernprozeß zur Verfügung stehenden Informationsangebot ist (cf., Satz 4.21).

Satz 4.30. $SMON^d\text{-}INF \setminus LIM\text{-}TXT \neq \emptyset$

Beweis: Wir zeigen die Behauptung mit Hilfe der indizierten Familie $\mathcal{L}$ aus dem Beweis von Satz 4.3, Aussage (1). Wie man leicht nachrechnet arbeitet die im Beweis von Hilfssatz A definierte IIM M dual schwach-monoton bei Informantbelehrung.

q.e.d.

Mit den folgenden Corollaren vervollständigen wir das Bild hinsichtlich der Leistungsfähigkeit der eingeführten Typen monotonen und dual monotonen Lernens in Abhängigkeit vom Informationsangebot.

Corollar 4. 31.

(1) $SMON^d\text{-}TXT \subset SMON^d\text{-}INF$

(2) $MON^d\text{-}TXT \subset MON^d\text{-}INF$

(3) $WMON^d\text{-}TXT \subset WMON^d\text{-}INF$

Beweis: Aussage (1) ergibt sich direkt aus Satz 4.30. Da nach Corollar 4.17 $SMON^d\text{-}TXT \subset MON^d\text{-}TXT \subset WMON^d\text{-}TXT$, folgen die Aussagen (2) und (3) ebenfalls unmittelbar aus Satz 4.30.

q.e.d.

Corollar 4. 32.

(1) $SMON^d\text{-}TXT \subset FIN\text{-}INF$

(2) $SMON^d\!-\!INF \quad \# \quad MON^d\!-\!TXT$

(3) $MON^d\!-\!INF \quad \# \quad LIM\!-\!TXT$

(4) $MON^d\!-\!INF \quad \# \quad WMON^d\!-\!TXT$

Beweis: Nach Satz 4.21 ist $SMON^d\!-\!TXT = FIN\!-\!TXT$ und somit ergibt sich (1) direkt aus Satz 4.3.

Zum Nachweis von (2) ist nur noch $MON^d\!-\!TXT \setminus SMON^d\!-\!INF \neq \emptyset$ zu zeigen, da die verbleibende Beziehung $SMON^d\!-\!INF \setminus MON^d\!-\!TXT \neq \emptyset$ nach Satz 4.30 klar ist. Wir definieren eine indizierte Familie über dem Alphabet $\{a,b\}$ wie folgt: Seien $L_0 = \{a\}^*$, $L_k = \{a^n \mid 1 \leq n \leq k\} \cup \{b^{k+1}\}$ für alle ≥ 1 sowie $L_{k,m} = L_k \cup \{a^m\}$ für alle $m,k \in \mathbb{N}+$ mit $m > k$. $\mathcal{L}$ sei dann die kanonische Aufzählung aller dieser Sprachen. Wir zeigen zunächst, daß $\mathcal{L}$ dual monoton bzgl. $\mathcal{L}$ gelernt werden kann.

Mögen $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ sein. Wir definieren jetzt informal eine IIM M. Solange $t_x^+ \subseteq \{a\}^*$ verifiziert wird, gibt M die kanonische Nummer von L_0 aus. Gilt $t_x^+ \not\subseteq \{a\}^*$, so verfügt M über ein Wort der Form b^{k+1}. Die Maschine testet dann, ob das längste bis dahin im Text gelesene Wort $w \in \{a\}^*$ eine Länge kleiner gleich k hat. Gilt $|w| > k$, so gibt M die kanonische Nummer von $L_{k,|w|}$ aus und wiederholt diese Hypothese in jedem nachfolgenden Lernschritt. Andernfalls wird die kanonische Nummer von L_k ausgeben und in jedem folgenden Schritt getestet, ob in t_x ein Wort $w \in \{a\}^*$ mit $|w| > k$ vorkommt. Ist das nicht der Fall, so wird die kanonische Nummer für L_k wiederholt. Wenn ja, so gibt M die kanonische Nummer von $L_{k,|w|}$ aus und wiederholt diese Hypothese in jedem weiteren Lernschritt.

Nach Konstruktion lernt M offensichtlich L bei Textbelehrung. Weiterhin führt M höchstens zwei Hypothesewechsel aus; nämlich von j_x für L_0 zu j_{x+z} für L_k und schließlich j_{x+z+y} für $L_{k,m}$. Da $co\!-\!L_{k,m} = co\!-\!L_k \setminus \{a^m\}$ gilt $co\!-\!L_k \cap co\!-\!L_{k,m} = co\!-\!L_{k,m}$. Somit erhalten wir:

$$co\!-\!L_0 \cap co\!-\!L_{k,m} \subseteq co\!-\!L_k \cap co\!-\!L_{k,m} = co\!-\!L_{k,m},$$

womit M im Falle zweier Hypothesenwechsel dual monoton arbeitet. Die verbleibenden Fälle sind trivial. Damit ist $\mathcal{L} \subseteq MON^d\!-\!TXT(M)$.

Mit der gleichen Technik wie im Beweis von Hilfssatz 2 von Satz 4.16 zeigt man problemlos $\mathcal{L} \notin MON\!-\!INF$, womit sich wegen $SMON^d\!-\!INF \subset MON\!-\!INF$ (cf., Satz 4.26) sofort (2) ergibt.

q.e.d.

Wir beenden diesen Abschnitt, indem wir die bislang erhaltenen Resultate in Bildform zusammenfassen. Dabei beschränken wir uns wieder auf den klassentreuen Fall, da er schon die wesentlichen Eigenschaften der betrachteten Sprachlerntypen hinreichend gut widerspielgelt.

Monotones Lernen bei Textbelehrung versus
monotones Lernen bei Informantbehrung

$$FIN{-}TXT \subset SMON{-}TXT \subset MON{-}TXT \subset WMON{-}TXT \subset LIM{-}TXT$$

$$\cap \qquad \cap \qquad \cap \qquad \cap \qquad \cap$$

$$FIN{-}INF \subset SMON{-}INF \subset MON{-}INF \subset WMON{-}INF = LIM{-}INF$$

Abbildung 10

Das folgende Bild zeigt die Gemeinsamkeiten und Unterschiede zwischen monotoner und dual monotoner Inferenz.

Dual monotones Lernen bei Textbelehrung versus
dual monotones Lernen bei Informantbehrung

$$FIN{-}TXT = SMON^{d}{-}TXT \subset MON^{d}{-}TXT \subset WMON^{d}{-}TXT \subseteq LIM{-}TXT$$

$$\cap \qquad \cap \qquad \cap \qquad \cap \qquad \cap$$

$$FIN{-}INF \subset SMON^{d}{-}INF \subset MON^{d}{-}INF \subset WMON^{d}{-}INF = LIM{-}INF$$

Abbildung 11

4.3. Charakterisierungen

In diesem Abschnitt charakterisieren wir Typen des exakten und klassentreuen monotonen und dual monotonen Lernens. Wir diskutieren den klassenumfassenden Fall und stellen einige Resultate vor. Charakterisierungen spielen in allen Bereichen der algorithmischen Lernentheorie eine bedeutende Rolle. Sie gestatten oftmals einen wesentlich tieferen Einblick in die Art und Weise, wie Lernalgorithmen arbeiten können (cf., Angluin, 1980a, Blum und Blum, 1975, Jain und Sharma, 1989, Wiehagen, 1977, 1991, Zeugmann, 1983).

Die Geschichte der Charakterisierungen von Lerntypen begann mit der fundamentalen Arbeit Blum und Blum (1975). Seitdem wurden unterschiedlichste theoretische Konzepte zur Charakterisierung genutzt. Dabei kamen komplexitätstheoretische Methoden (cf., Blum und Blum, 1975, Wiehagen und Liepe, 1976, Zeugmann, 1983) genauso zum Tragen wie numerierungstheoretische Termini (cf., Wiehagen, 1991 und die dortigen Verweise). Generell kann festgestellt werden, daß Charakterisierungen häufig grundlegende Einsichten in die Struktur derjenigen Objekte gestatten, die unter dem charakterisierten Paradigma erlernbar sind. Ein illustratives Beispiel hierfür ist Angluins (1980a) Charakterisierung der exakt bei Textbelehrung erlernbaren indizierten Familien. Insbesondere gestattet dieser Satz Einsichten dazu, wie eine IIM übergeneralisierte Hypothesen erkennen und wechseln kann.

Satz 4.33. (Angluin, 1980a)
Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in LIM-TXT$ genau dann, wenn es eine effektive Prozedur gibt, die bei jeder Eingabe $j \in \mathbb{N}$ eine endliche Menge T_j von Wörtern rekursiv aufzählt, die folgenden Bedingungen genügt:

(1) *T_j ist endlich,*

(2) *$T_j \subseteq L_j$ für alle $j \in \mathbb{N}$,*

(3) *für alle $j, z \in \mathbb{N}$: $T_j \subseteq L_z$ impliziert $L_z \not\subset L_j$.*

Wenn eine uniform rekursiv aufzählbare Familie $(T_j)_{j \in \mathbb{N}}$ endlicher Mengen die Bedingungen (1) bis (3) erfüllt, so nennen wir die Mengen T_j ***Tell-tales***.

Ursprünglich charakterisierte dieser Satz lediglich exaktes Lernen im Limes. Nach Satz 4.1 gilt jedoch $ELIM-TXT = LIM-TXT = KLIM-TXT$. Somit charakterisiert Satz 4.33 Lernen im Limes vollständig.

Obwohl Angluin (1980a) einige hinreichende Bedingungen formulierte, die exaktes konservatives Lernen implizierten, blieb offen, ob sich konservatives Lernen auch in Termen von rekursiven Aufzählungen endlicher nicht leerer Mengen charakterisieren läßt. Diese Frage konnte von S. Lange gemeinsam mit dem Autor beantwortet werden (cf., Lange und Zeugmann, 1992c). Wir wollen diesen Abschnitt deshalb mit der Lösung dieses lange offen gebliebenen Problems beginnen. Nach den Sätzen aus Abschnitt 4.1 wissen wir bereits, daß die Wahl des Hypothesenraumes entscheidenen Einfluß auf die Leistungsfähigkeit konservativer IIMs haben kann. Wir stellen deshalb zunächst die Charakterisierung klassentreuen konservativen Lernens vor und diskutieren anschließend die verbleibenden Fälle.

4.3.1. Charakterisierung konservativer Inferenz

Bevor wir den ersten Satz dieses Abschnitts angeben sei noch vereinbart für jeden Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ mit $L(\mathcal{G})$ die Menge $\{L(G_j) | j \in \mathbb{N}\}$ zu bezeichnen. Wir definieren weiterhin die rekursive Generierbarkeit von Mengen wie folgt.

Definition 19. it ***Eine Familie*** $(\hat{T}_j)_{j\in\mathbb{N}}$ ***endlicher Mengen heißt rekursiv generierbar***, falls es eine rekursive Prozedur g gibt, die bei Eingabe von j alle Elemente von T_j generiert und dann stoppt. Endet die Berechnung von $g(j)$ ohne Ausgabe, so ist $T_j = \emptyset$.

Jetzt können wir die versprochene Charakterisierung vorstellen und beweisen.

Satz 4.34. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in$ CONSERVATIVE–TXT genau dann, wenn es einen Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ und eine rekursiv generierbare Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ endlicher nichtleerer Mengen derart gibt, daß*

(1) *Val $\mathcal{L} = L(\hat{\mathcal{G}})$,*

(2) *$\hat{T}_j \subseteq L(\hat{G}_j)$ für alle $j \in \mathbb{N}$,*

(3) *für alle j, $z \in \mathbb{N}$: $\hat{T}_j \subseteq L(\hat{G}_z)$ impliziert $L(\hat{G}_z) \not\subset L(\hat{G}_j)$.*

Beweis: Notwendigkeit. Sei $\mathcal{L} \in$ *CONSERVATIVE–TXT*. Dann gibt es einen Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ und eine IIM M, so daß $\mathcal{L} \subseteq$ *CONSERVATIVE–TXT*(M) bzgl. $\mathcal{G}$. Wir zeigen zunächst, wie der Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ konstruiert wird. Die Konstruktion wird in zwei Schritten ausgeführt. Wir geben erst einen Hypothesenraum $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ und eine rekursiv generierbare Familie $(\tilde{T}_j)_{j\in\mathbb{N}}$ von endlichen, jedoch möglicherweise leeren Mengen an. Der gesuchte Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ wird danach aus $\tilde{\mathcal{G}}$ gewonnen. Möge $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ die Cantorsche Funktion zur Kodierung aller Paare natürlicher Zahlen sein. Den Raum $\tilde{\mathcal{G}}$ und die Familie $(\tilde{T}_j)_{j\in\mathbb{N}}$ definieren wir folgendermaßen. Bei Eingabe von j werden zunächst die Zahlen $k, x \in \mathbb{N}$ berechnet, für die $j = c(k, x)$ ist. Wir setzen dann $\tilde{G}_{c(k,x)} = G_k$ und definieren

$$\tilde{T}_{c(k,x)} = \begin{cases} Val\ t^k_y, & \text{falls} \quad y = min\{z | z \leq x,\ M(t^k_z) = k\}, \\ \emptyset, & \text{sonst}, \end{cases}$$

wobei t^k den kanonischen Text von $L(G_k)$ bezeichnet. Offensichtlich sind die Mengen $\tilde{T}_{c(k,x)}$ uniform rekursiv generierbar und endlich. Der gewünschte Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ entsteht aus $\tilde{\mathcal{G}}$, indem alle Grammatiken $\tilde{G}_{c(k,x)}$ gestrichen werden, für die

$\tilde{T}_{c(k,x)} = \emptyset$. Die Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ erhält man analog durch Streichen aller leeren $\tilde{T}_{c(k,x)}$. Damit ist $(\hat{T}_j)_{j\in\mathbb{N}}$ eine rekursiv generierbare Familie endlicher und nichtleerer Mengen. Um unnötigen Bezeichnungsaufwand zu vermeiden nehmen wir auf $\hat{T}_j$ durch $\hat{T}_{c(k,x)}$ Bezug, d.h. wir verzichten auf die explizite Benennung der Abbildung der nichtleeren $\tilde{T}_j$ aus $(\tilde{T}_j)_{j\in\mathbb{N}}$ auf $(\hat{T}_j)_{j\in\mathbb{N}}$. Wir haben zu zeigen, daß die Bedingungen (1) bis (3) erfüllt sind.

Aus der Konstruktion ergibt sich (2) unmittelbar. Um (1) zu beweisen, sei $L \in \mathcal{L}$. Es muß gezeigt werden, daß mindestens ein j mit $j = c(k,x)$ und $L = L(\hat{G}_{c(k,x)})$ existiert. Dafür genügt nun der Nachweis von $\tilde{T}_{c(k,x)} \neq \emptyset$. Möge t^L der kanonische Text für L sein. Da M die Sprache L insbesondere auch auf t^L zu inferieren hat, existieren Zahlen k, x so daß $M(t^L_x) = k$ und $M(t^L_z) \neq k$ für alle $z < x$ ist. Damit ist $\tilde{T}_{c(k,x)} = Val\ t^k_x$. Somit ist auch $\hat{T}_{c(k,x)} \neq \emptyset$. Weiterhin folgt aus der Konstruktion unmittelbar $L = L(\hat{G}_{c(k,y)})$ für alle $y \geq x$, da M nach Voraussetzung konservativ arbeitet. Damit ist (1) bewiesen.

Eigenschaft (3) zeigen wir indirekt. Angenommen, es gibt $j,\ z \in \mathbb{N}$ so daß $\hat{T}_j \subseteq L(\hat{G}_z)$ and $L(\hat{G}_z) \subset L(\hat{G}_j)$. Nach der Definition der Tell-tale Mengen gibt es dann eindeutig bestimmte Zahlen $k, x \in \mathbb{N}$ mit $j = c(k,x)$. Seien jetzt $w_1, ..., w_y$ die Wörter von $\hat{T}_j$ in kanonischer Ordnung bzgl. $L(\hat{G}_{c(k,x)})$. Nach Konstruktion ist dann $M(w_1, ..., w_y) = k$. Da $\hat{T}_j \subseteq L(\hat{G}_z) \subset L(\hat{G}_j) = L(\hat{G}_{c(k,x)})$, können wir schließen, daß $w_1, ..., w_y$ ein Anfangsstück des kanonischen Textes für $L(\hat{G}_z)$ ist. Da M auch $L(\hat{G}_z)$ auf ihrem kanonischen Text lernt, muß auf jeder Fortsetzung von $w_1, ..., w_y$ zu einem Text für $L(\hat{G}_z)$ mindestens ein Hypothesenwechsel auftreten, der offensichtlich nicht durch Inkonsistenz bedingt sein kann. Dieser Widerspruch beweist (3).

Hinlänglichkeit. Es genügt eine IIM M zu beschreiben, die jede Sprache $L \in \mathcal{L}$ bei Textbelehrung bzgl. $\hat{\mathcal{G}}$ konservativ lernt. Seien also $L \in \mathcal{L}$ und $t \in Text(L)$ sowie $x \in \mathbb{N}$. Wir definieren:

$M(t_x) =$ „Wenn $x = 0$ oder $x > 0$ und M nach dem Lesen von t_{x-1} noch keine Hypothese ausgegeben hat, so führe (B) aus. Andernfalls arbeite (A) ab.

(A) Sei j die letzte von M nach dem Lesen von t_{x-1} produzierte Hypothese. Teste, ob $t^+_x \subseteq L(\hat{G}_j)$. Wenn ja, so gib j aus und lies das nächste Wort auf dem Eingabeband. Andernfalls arbeite (B) ab.

(B) Generiere $\hat{T}_j$ für $j = 1, ..., x$ und teste, ob $\hat{T}_j \subseteq t^+_x \subseteq L(\hat{G}_j)$. Wird mindestens ein j gefunden, das den Test erfüllt, so gib das kleinste derartige j aus und lies das nächste Wort auf dem Eingabeband. Andernfalls gib keine Hypothese aus und lies das nächste Wort auf dem Eingabeband.“

Da alle $\hat{T}_j$ uniform rekursiv generierbar und endlich sind, ist M eine IIM. Nach Konstruktion ändert M ihre Hypothese nur dann, wenn in (A) eine Inkonsistenz gefunden wird. Zum Nachweis der Konservativität von M haben wir noch zu zeigen, daß M auf t gegen eine für L korrekte Hypothese konvergiert.

***Hilfssatz* 1**. M konvergiert auf t.

Sei $z = \mu k[L = L(\hat{G}_k)]$. Wir betrachten $\hat{T}_1$, ..., $\hat{T}_z$. Nach der Wahl von z muß ein kleinstes x mit $\hat{T}_z \subseteq t_x^+ \subseteq L(\hat{G}_z)$ existieren. Somit gibt M spätestens nach dem Lesen von t_x eine Hypothese aus und auch in jedem nachfolgenden Schritt. Weiterhin gilt für jede nach dem Lesen von t_x produzierte Hypothese j nach Konstruktion $j \leq x$, da $z \in \{k \leq x \mid \hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)\}$. Nun wechselt M ihre Hypothese nur im Fall einer gefundenen Inkonsistenz, womit sofort klar ist, daß eine einmal verworfene Hypothese nie wieder in Betracht gezogen wird. Da zumindest die Hypothese z nie verworfen werden kann, muß M folglich konvergieren.

q.e.d. (Hilfssatz 1)

***Hilfssatz* 2**. Wenn M konvergiert, sagen wir gegen j, so ist $L = L(\hat{G}_j)$.

Angenommen nicht; dann konvergiert M gegen j und es gilt $L \neq L(\hat{G}_j)$.
Fall 1. $L \setminus L(\hat{G}_j) \neq \emptyset$
Folglich gibt es mindestens ein Wort $w \in L \setminus L(\hat{G}_j)$. Dieses Wort muß aber irgendwann im Text erscheinen, sagen wir an der Stelle r. Somit verwirft M ihre Hypothese j spätestens, nachdem t_r gelesen wurde, und kann damit nicht mehr gegen j konvergieren.
Fall 2. $L(\hat{G}_j) \setminus L \neq \emptyset$
Dann können wir o.B.d.A. den Fall $L \subset L(\hat{G}_j)$ betrachten, da wir andernfalls wieder in Fall 1 wären. Nach Konstruktion von M existiert nun ein $r \in \mathbb{N}$, so daß M in Anweisung (B) die Gültigkeit von $\hat{T}_j \subseteq t_r^+ \subseteq L(\hat{G}_j)$ verifiziert, da andernfalls j nie als Hypothese in Frage käme. Sei wieder $z = \mu k[L = L(\hat{G}_k)]$. Da $L = L(\hat{G}_z)$ und $t_r^+ \subseteq L(\hat{G}_z)$ aber für jedes $r \in \mathbb{N}$ erfüllt ist, ergibt sich somit $\hat{T}_j \subseteq L(\hat{G}_z)$. Mit Eigenschaft (3) erhalten wir $L = L(\hat{G}_z) \not\subset L(\hat{G}_j)$, im Widerspruch zu unserer Annahme. Also wird jedes $L \in \mathcal{L}$ von M bei Textbelehrung gelernt.

q.e.d.

Wir merken noch an, daß die Lösung des Problems, klassentreues konservatives Lernen zu charakterisieren, unabhängig von uns, auch Bilardi und Kapur (1992) gelungen ist. Allerdings unterscheidet sich ihr konzeptionelles Vorgehen deutlich vom oben präsentierten Herangehen. Um das deutlich zu machen, brauchen wir folgende Notation. Sei A eine endliche Menge und $\mathcal{L}$ eine indizierte Familie. Eine Sprache $L \in \mathcal{L}$ heißt kleinste obere Schranke für A, falls $A \subseteq L$ und für alle $\hat{L} \in \mathcal{L}$ gilt: Wenn $A \subseteq \hat{L}$, so ist $\hat{L}$ keine

echte Teilmenge von L. Kapur und Bilardi (1992) zeigten nun, daß konservatives Lernen äquivalent zur Existenz einer rekursiven Aufzählung von Paaren aus endlichen Mengen und Grammatiken ist, so daß jede Grammatik eine Sprache generiert, die kleinste obere Schranke für die im Paar stehende endliche Menge ist. Somit basiert Kapur und Bilardis (1992) Beweis konzeptionell auf der geschickten Anwendung einer Funktion, die zu endlichen Mengen kleinste obere Schranken berechnet. Der vor uns vorgestellte Ansatz ist im wesentlichen dual dazu, da wir zunächst einen Hypothesenraum gebaut haben und anschließend die Tell-tale Mengen rekursiv berechnet wurden.

Allerdings stellt der obige Satz noch nicht ganz die Lösung des Angluinschen (1980a) Problems dar, da sie sich ausschließlich mit exaktem Lernen beschäftigte. Es macht allerdings keine Mühe, den Beweis von Satz 4.34 so zu modifizieren, daß wir eine Charakterisierung exakten konservativen Lernens erhalten.

Satz 4.35. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in ECONSERVATIVE\text{–}TXT$ genau dann, wenn es eine rekursiv generierbare Familie $(T_j^y)_{j,y\in\mathbb{N}}$ endlicher Mengen derart gibt, daß*

(1) *für alle $L \in \mathcal{L}$ existiert ein j mit $L_j = L$ und $T_j^y \neq \emptyset$ für fast alle $y \in \mathbb{N}$,*

(2) *für alle $j, y \in \mathbb{N}$ gilt: $T_j^y \subseteq L_j$ und $T_j^y \neq \emptyset$ impliziert $T_j^y = T_j^{y+1}$,*

(3) *für alle $j,\ y,\ z \in \mathbb{N}$ gilt: $\emptyset \neq T_j^y \subseteq L_z$ impliziert $L_z \not\subset L_j$.*

Beweis: Der Beweis erfolgt genau wie oben, wobei anstelle des Hypothesenraumes $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ unmittelbar die indizierte Familie $\mathcal{L}$ verwendet wird. Wenn eine Sprache L nun mehrfach in der Aufzählung vorkommt, so muß nicht jede Nummer k von L auch als Hypothese auf dem kanonischen Text von L ausgegeben werden. Somit können in $(T_j^y)_{j,y\in\mathbb{N}}$ auch leere Mengen aufgezählt werden. Da L jedoch gelernt wird, gibt es stets ein j mit $L = L_j$, so daß $T_j^y \neq \emptyset$ für fast alle $y \in \mathbb{N}$. Mutatis mutandis läßt sich alles weitere in völliger Analogie zum Beweis von Satz 4.34 zeigen.

q.e.d.

Wir betrachten abschließend in diesem Abschnitt den klassenumfassenden Fall. Zumindest aus beweistechnischer Sicht ergeben sich jetzt neue Probleme. Die Ursache hierfür ist darin zu sehen, daß wir a priori nicht sagen können, wie sich eine IIM auf beliebigen Texten einer nicht zur indizierten Familie gehörenden Sprache verhält. Insbesondere muß dort nicht konservativ gearbeitet werden. Der Beweis des folgenden Satzes zeigt, wie man diese Schwierigkeiten überwinden kann.

Satz 4.36. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in KCONSERVATIVE\text{–}TXT$*

genau dann, wenn es einen Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ *und eine rekursiv generierbare Familie* $(\hat{T}_j)_{j\in\mathbb{N}}$ *endlicher nichtleerer Mengen derart gibt, daß*

(1) $Val\ \mathcal{L} \subseteq L(\hat{\mathcal{G}})$,

(2) $\hat{T}_j \subseteq L(\hat{G}_j)$ *für alle* $j \in \mathbb{N}$,

(3) *für alle* $j,\ z \in \mathbb{N}$*:* $\hat{T}_j \subseteq L(\hat{G}_z)$ *impliziert* $L(\hat{G}_z) \not\subset L(\hat{G}_j)$.

Beweis: Notwendigkeit. Möge $\mathcal{L} \in KCONSERVATIVE\text{–}TXT$ sein. Dann existieren ein klassenumfassender Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ sowie eine konsistent arbeitende IIM M derart, daß $\mathcal{L} \subseteq KCONSERVATIVE\text{–}TXT(M)$ bzgl. $\mathcal{G}$ (cf., Satz 4.18). Wir definieren zunächst genauso wie im Beweis von Satz 4.34 einen Hypothesenraum $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$. Zur Konstruktion der Familie $(\tilde{T}_j)_{j\in\mathbb{N}}$ sind jedoch wichtige Modifikationen nötig, die wir im folgenden beschreiben. Für $k, r \in \mathbb{N}$ mögen t^k bzw. τ^r den kanonischen Text von $L(G_k)$ bzw. L_r bezeichnen. Dann definieren wir $\tilde{T}_j$ für alle $j \in \mathbb{N}$ wie folgt.

(A1) Berechne $k, x \in \mathbb{N}$ mit $j = c(k, x)$.

(A2) Teste, ob ein $r \leq x$ derart existiert, daß $t^k_x = \tau^r_x$. Ist der Test für kein $r \leq x$ erfüllt, so setzen wir $\tilde{T}_j = \emptyset$. Andernfalls wird (A3) abgearbeitet.

(A3) Setze

$$\tilde{T}_{c(k,x)} = \begin{cases} Val\ t^k_y, & \text{falls} \quad y = min\{z \mid z \leq x,\ M(t^k_z) = k\}, \\ \emptyset, & \text{sonst.} \end{cases}$$

Wie schon im Beweis von Satz 4.34 gezeigt wurde, sind die Anweisungen (A1) und (A3) effektiv ausführbar. Weiterhin terminiert die Abarbeitung dieser Anweisungen stets. Der Test in Anweisung (A2) kann effektiv ausgeführt werden, da $\mathcal{L}$ eine indizierte Familie ist. Offensichtlich terminiert auch die Abarbeitung von (A2). Somit sind die Mengen $\tilde{T}_j$ für alle j rekursiv generierbar und nach Konstruktion endlich.

Die gewünschte Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ erhält man aus $(\tilde{T}_j)_{j\in\mathbb{N}}$ durch Streichen aller $\tilde{T}_j$ mit $\tilde{T}_j = \emptyset$. Analog wird $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ aus $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ gewonnen. Wir verzichten wieder auf die explizite Angabe der entsprechenden Abbildung und nehmen auf $\hat{T}_j$ auch durch $\hat{T}_{c(k,x)}$ Bezug.

Wir haben jetzt zu zeigen, daß die Eigenschaften (1) bis (3) erfüllt sind. Nach Konstruktion ist (2) offensichtlich und (1) kann mit den gleichen Mitteln wie im Beweis von Satz 4.34 gezeigt werden. Eigenschaft (3) beweisen wir indirekt. Angenommen, es gibt

Indizes $j, z \in \mathbb{N}$ mit $\hat{T}_j \subseteq L(\hat{G}_z)$ und $L(\hat{G}_z) \subset L(\hat{G}_j)$. Mögen dann $k, m, x, y \in \mathbb{N}$ Zahlen derart sein, daß $j = c(k, x)$ und $z = c(m, y)$. Weiterhin seien $w_0, ..., w_n$ die Wörter aus $\hat{T}_j$ geschrieben als Anfangsstück des kanonischen Textes von $L(\hat{G}_j)$ bis zur minimalen Länge $n + 1$, für die zum ersten Mal $M(w_0, ..., w_n) = k$ gilt. Nach Konstruktion ist dann $L(\hat{G}_j) = L(G_k)$. Die zu überwindene Schwierigkeit besteht darin, daß wir nicht $L(G_k) \in \mathcal{L}$ schließen können. In diesem Fall ließe sich mühelos wie im Beweis von Satz 4.34 ein Widerspruch konstruieren. Ist jedoch $L(G_k) \notin \mathcal{L}$, so wissen wir a priori nichts. Mit dem folgenden Hilfssatz erweitern wir unsere Kenntnisse in ausreichender Weise.

***Hilfssatz* 1**. Für alle $u < n$ gilt $M(w_0, ..., w_u) \neq m$.

Nach Voraussetzung ist $\hat{T}_j \subseteq L(\hat{G}_z)$ und nach Konstruktion gilt $L(\hat{G}_z) = L(G_m)$. Damit ist $w_0, ..., w_u$ für alle $u < n$ Anfangsstück eines Textes von $L(\hat{G}_z)$. Wäre nun $M(w_0, ..., w_u) = m$ für ein $u < n$, so müßte $\{w_0, ..., w_n\} \not\subseteq \hat{T}_j$ gelten, im Widerspruch zu $\hat{T}_j = \{w_0, ..., w_n\}$. Um das zu sehen, erinnern wir daran, daß $\hat{T}_j \neq \emptyset$. Somit stellt $w_0, ..., w_n$ nach Konstruktion auch ein Anfangsstück des kanonischen Textes einer Sprache $L \in \mathcal{L}$ dar. Folglich muß M auf *diesem* Anfangsstück sowohl *konservativ* als auch *konsisitent* arbeiten. Wechselte die Maschine M auf $w_0, ..., w_n$ in ihrer bis zum Lesen von w_n produzierten Hypothesenfolge von m zu k (nicht notwendigerweise in einem Schritt), so müßte für jeden Wechsel eine Inkonsistenz bzgl. der schon gelesenen Wörter vorliegen. Sei also $m = M(w_0, ..., w_u)$. Nach Konstruktion wissen wir weiterhin, daß $k = M(w_0, ..., w_n)$. Da M konsistent arbeitet, erhalten wir $\{w_0, ..., w_u\} \subseteq L(G_m)$ und $\{w_0, ..., w_n\} \subseteq L(G_k)$. Aus der Konservativität von M folgt weiterhin $\{w_0, ..., w_n\} \not\subseteq L(G_m)$. Nun ist $L(\hat{G}_z) = L(G_m)$ und nach Voraussetzung $\hat{T}_j \subseteq L(\hat{G}_z)$. Also gilt erst recht $\{w_0, ..., w_n\} \not\subseteq \hat{T}_j$.

q.e.d. (Hilfssatz 1)

Wir beenden den Beweis von Eigenschaft (3) durch folgenden Hilfssatz.

***Hilfssatz* 2**. Wenn $L(\hat{G}_z) \subset L(\hat{G}_j)$, so gibt es ein Anfangsstück eines Textes einer Sprache $L \in \mathcal{L}$, auf dem M nicht konservativ arbeitet.

Da $\hat{T}_j \subseteq L(\hat{G}_z) \subset L(\hat{G}_j)$ und M auf $w_0, ..., w_n$ nach Hilfssatz 1 nicht m ausgibt, muß $w_0, ..., w_n$ auch Anfangsstück des kanonischen Textes von $L(\hat{G}_z)$ sein. Nach Konstruktion hat der kanonische Text τ^r einer Sprache $L_r \in \mathcal{L}$ mit dem kanonischen Text t^m von $L(G_m) = L(\hat{G}_z)$ bis zur Länge $y > n$ übereinzustimmen, da $z = c(m, y)$. Möge $w_0, ..., w_n, w_{n+1}, ..., w_y$ dieses Anfangsstück bezeichnen. Aus der Konstruktion ergibt sich weiterhin die Existenz einer Zahl $s \leq y$, so daß $M(w_0, ..., w_s) = m$. Nach Hilfssatz 1 ist $s > n$. Damit haben wir ein Anfangsstück eines Textes einer Sprache $L_r \in \mathcal{L}$ auf dem M irgendwann die Hypothese k (auf $w_0, ..., w_n$ nämlich) ausgibt und nach dem Lesen

weiterer Worte dieses Anfangsstückes m als Hypothese produziert (auf $w_0, ..., w_s$). Da $L(G_k) = L(\hat{G}_j) \supset L(\hat{G}_z) = L(G_m)$ ist k aber eine für $L(G_m)$ übergeneralisierte Hypothese. Folglich kann der Wechsel von k zu m nicht durch Inkonsistenz mit dem bis dahin gelesenem Text erzwungen worden sein. Somit arbeitet M auf diesem Anfangsstück eines Textes der Sprache L_r nicht konservativ.

q.e.d. (Hilfssatz 2)

Damit ist die Notwendigkeit der Eigenschaften (1) bis (3) gezeigt.

Hinlänglichkeit. Die Hinlänglichkeit zeigt man in völliger Analogie zum Beweis von Satz 4.34.

q.e.d.

4.3.2. Charakterisierung streng-monotonen Lernens

Die im letzten Abschnitt entwickelte Beweistechnik läßt sich jetzt mit geringfügigen Modifiakationen direkt auf den streng-monotonen exakten und klassentreuen Fall anwenden.

Satz 4.37. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in SMON-TXT$ genau dann, wenn es einen Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ und eine rekursiv generierbare Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ endlicher nichtleerer Mengen derart gibt, daß*

(1) *$Val\ \mathcal{L} = L(\hat{\mathcal{G}})$,*

(2) *$\hat{T}_j \subseteq L(\hat{G}_j)$ für alle $j \in \mathbb{N}$,*

(3) *für alle $j,\ z \in \mathbb{N}$: $\hat{T}_j \subseteq L(\hat{G}_z)$ impliziert $L(\hat{G}_j) \subseteq L(\hat{G}_z)$.*

Beweis: Notwendigkeit. Sei $\mathcal{L} \subseteq SMON-TXT(M)$ für eine IIM M und einen klassentreuen Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$. Die rekursiv generierbare Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ wird genauso wie im Beweis von Satz 4.34 definiert. Völlig analog beweist man nun die Gültigkeit von (1) und (2). Zum Nachweis von (3) sei $\hat{T}_j \subseteq L(\hat{G}_z)$. Wir haben $L(\hat{G}_j) \subseteq L(\hat{G}_z)$ zu zeigen. Mögen k, x die eindeutig bestimmten Zahlen mit $j = c(k, x)$ sein. Weiterhin seien $w_0, ..., w_y$ die Wörter aus $\hat{T}_j$ in kanonischer Ordnung bzgl. $L(\hat{G}_{c(k,x)})$ bis zur minimalen Länge $y + 1$ für die zum ersten Mal $k = M(w_0, ..., w_y)$. Da $\hat{T}_j \subseteq L(\hat{G}_z)$, sehen wir, daß $w_0, ..., w_y$ auch Anfangsstück eines Textes für $L(\hat{G}_z)$ ist. Somit kann $w_0, ..., w_y$ zu einem Text für $L(\hat{G}_z)$ fortgesetzt werden. Auf diesem Text muß M streng-monoton arbeiten und $L(\hat{G}_z)$ lernen. Also gibt M auf diesem Text nach dem Lesen von $w_0, ..., w_y$ die Hypothese k und in einem späteren Lernschritt eine für $L(\hat{G}_z)$ korrekte Hypothese n aus. Aus der strengen Monotonie folgt wegen der Transitivität von „$\subseteq$“ somit $L(G_k) \subseteq L(G_n)$. Da

n für $L(\hat{G}_z)$ korrekt ist, gilt weiterhin $L(\hat{G}_z) = L(G_n)$ und nach Konstruktion überdies $L(G_k) = L(\hat{G}_j)$. Also erhalten wir $L(\hat{G}_j) \subseteq L(\hat{G}_z)$.

Hinlänglichkeit. Es genügt zu zeigen, daß $\mathcal{L}$ bzgl. des Hypothesenraumes $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in \mathbb{N}}$ streng-monoton inferiert werden kann. Mögen $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$ sein. Die gewünschte IIM M definieren wir wie folgt:

$M(t_x) =$ „Generiere für $j = 1, ..., x$ die Mengen $\hat{T}_j$ und teste, ob $\hat{T}_j \subseteq t_x^+ \subseteq L(\hat{G}_j)$. Wird wenigstens ein j gefunden, das den Test erfüllt, so gib das kleinste derartige j aus und lies anschließend die nächste Eingabe.

Andernfalls gib keine Hypothese aus und lies die nächste Eingabe."

Wir haben noch zu zeigen, daß M das Verlangte leistet.

***Hilfssatz* 1**. M lernt L

Sei $k = \mu z[L = L(\hat{G}_z)]$. Dann konvergiert M gegen k. Um das zu sehen betrachten wir $\hat{T}_0, ..., \hat{T}_k$. Für hinreichend große x gilt somit $\hat{T}_k \subseteq t_x^+ \subseteq L(\hat{G}_k)$. Damit muß M, nachdem t_x gelesen wurde, in jedem weiteren Schritt eine Hypothese $j \leq k$ ausgeben. Aus der Konstruktion folgt weiterhin, daß eine einmal verworfene Hypothese auch in keinem weiteren Lernschritt ausgegeben wird. Angenommen, M konvergiert gegen eine Hypothese j mit $j < k$. Nach der Wahl von k gilt dann $L(\hat{G}_j) \neq L(\hat{G}_k)$.

Fall 1. $L \subset L(\hat{G}_j)$

Nach der Definition von M existiert dann ein n mit $\hat{T}_j \subseteq t_n^+ \subseteq L(\hat{G}_j)$. Da $t \in \mathit{Text}(L)$ und $L = L(\hat{G}_k)$, erhalten wir $\hat{T}_j \subseteq L(\hat{G}_k)$, woraus sich nach (3) sofort $L(\hat{G}_j) \subseteq L(\hat{G}_k)$ ergibt, im Widerspruch zu $L \subset L(\hat{G}_j)$.

Fall 2. $L \setminus L(\hat{G}_j) \neq \emptyset$

Sei $w \in L \setminus L(\hat{G}_j)$. Dann muß w irgendwann im Text t erscheinen, etwa an der Stelle $n + 1$. Für alle $x \geq n$ gilt somit $t_x^+ \not\subseteq L(\hat{G}_j)$, womit für kein $x \geq n$ die Hypothese j ausgegeben werden kann. Also konvergiert M gegen k und lernt somit L.

q.e.d. (Hilfssatz 1)

***Hilfssatz* 2**. M arbeitet streng-monoton.

Möge M irgendwann die Hypothese j und in einem späteren Schritt die Hypothese z ausgeben. Folglich existieren Zahlen n, m mit $n < m$ derart, daß $\hat{T}_j \subseteq t_n^+ \subseteq L(\hat{G}_j)$ und $\hat{T}_z \subseteq t_m^+ \subseteq L(\hat{G}_z)$. Damit erhalten wir sofort $\hat{T}_j \subseteq L(\hat{G}_z)$, woraus nach (3) die Beziehung $L(\hat{G}_j) \subseteq L(\hat{G}_z)$ folgt. Also arbeitet M streng-monoton.

q.e.d.

Es macht nun keine Mühe, auch exaktes streng-monotones Lernen mit den gleichen Modifikationen wie im Beweis von Satz 4.35 zu charakterisieren. Die Ausdehnung dieser Techniken auf den klassenumfassenden Fall steht hingegen noch aus. Dabei sind die Schwierigkeiten inhaltlich von gleicher Natur, wie die unmittelbar vor Satz 4.36 geschilderten Probleme.

Wir wollen dennoch eine Charakterisierung klasssenumfasssenden streng-monotonen Lernens angeben, die zu interessanten Einsichten in die Arbeitsweise streng-monoton arbeitender IIMs führt. Inhaltlich geht es dabei um die Frage, inwieweit der Lernprozeß durch iterativ arbeitende IIMs realisierbar ist. Iteratives Lernen wurde von Wiehagen (1976) eingeführt und seitdem intensiv studiert (cf., z.B. Jantke und Beick, 1981, Porat und Feldman, 1988). Beim iterativen Lernen wird die aus der Sicht potentieller Anwendungen unrealistische Forderung, einer IIM in jedem Lernschritt sukzessiv wachsende Anfangstücke von Informationsfolgen zur Verfügung zu stellen, durch die Bedingung ersetzt, daß eine IIM in jedem Schritt nur ihre letzte Hypothese und ein weiteres Wort der Informationsfolge verwenden darf. Formal erhalten wir folgende Definition.

Definition 20. *Sei $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum.* ***$\mathcal{L}$ heißt iterativ bei Textbelehrung bzgl. $\mathcal{G}$ erlernbar*** *(Bez. $\mathcal{L} \in IT{-}TXT$), falls es eine IIM M gibt, so daß für alle $L \in \mathcal{L}$ und alle $t \in Text(L)$ gilt:*

(1) *für alle $n \in \mathbb{N}$ sind $M_n(t)$ definiert, wobei $M_0(t) =_{df} M(t_0)$ und $M_{n+1}(t) =_{df} M(M_n(t), w_{n+1})$*

(2) *die Folge $(M_n(t))_{n\in\mathbb{N}}$ konvergiert im Limes gegen eine Zahl j mit $L = L(G_j)$.*

Damit arbeiten iterative IIMs wesentlich effizienter hinsichtlich ihres Bedarfs an Speicherplatz. Wir merken noch an, daß sich aus der Forderung, iterativ arbeiten zu müssen, unmittelbar ableitet, daß in jedem Schritt eine Hypothese zu produzieren ist.

Mit dieser Begriffsbildung können wir klassenumfassendes streng-monotones Lernen wie folgt charakterisieren.

Satz 4.38. $KSMON{-}TXT = KSMON{-}IT{-}TXT$

Beweis: Da die Inklusion $KSMON{-}IT{-}TXT \subseteq KSMON{-}TXT$ offensichtlich gilt, genügt es, $KSMON{-}TXT \subseteq KSMON{-}IT{-}TXT$ zu zeigen. Möge $\mathcal{L} \in KSMON{-}TXT$ sein. Dann gibt es eine IIM M und einen klassenumfassenden Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ derart, daß $\mathcal{L} \subseteq KSMON{-}TXT(M)$ bzgl. $\mathcal{G}$. Da wir im klassenumfassenden Fall sind können wir o.B.d.A. weiterhin voraussetzen, daß M auf jedem Text einer Sprache über dem zugrunde liegenden Alphabet in jedem Lernschritt eine Hypothese ausgibt und

zusätzlich konsistent arbeitet.

Wir definieren jetzt eine IIM $\tilde{M}$ sowie einen Hypothesenraum $\tilde{\mathcal{G}} = (\tilde{G}_j)_{j\in\mathbb{N}}$ und zeigen anschließend, daß $\tilde{M}$ die Familie $\mathcal{L}$ streng-monoton und iterativ bzgl. $\tilde{\mathcal{G}}$ lernt. Der Hypothesenraum $\tilde{\mathcal{G}}$ entsteht aus $\mathcal{G}$ durch kanonische Aufzählung des Abschlusses von $\mathcal{G}$ bzgl. endlicher Vereinigungen. Da das Wortproblem bzgl. aller Sprachen $L(G_j)$ uniform entscheidbar ist, erhält man leicht eine Prozedur zur uniformen Entscheidung des Wortproblems für alle Sprachen $L(\tilde{G}_j)$ (cf., Bucher und Maurer, 1984). Die Maschine $\tilde{M}$ definieren wir in Stufen, wobei Stufe k konzeptionell die Arbeit von $\tilde{M}_k$ beschreibt. Mögen jetzt $L \in \mathcal{L}$ und $t \in \mathit{Text}(L)$ sein. Wir setzen:

Stufe 0: Sei $t_0 = s_0$. Berechne $j_0 := M(s_0)$. Gib j_0 aus und beginne Stufe 1.

Stufe k: $\tilde{M}$ erhält als Eingabe j_{k-1} und das $(k+1)$–ste Element s_k von t. Teste, ob $s_k \in L(\tilde{G}_{j_{k-1}})$.

Fall 1. $s_k \in L(\tilde{G}_{j_{k-1}})$.
Setze $j_k = j_{k-1}$, gib j_k aus und beginne Stufe $k+1$.

Fall 2. $s_k \notin L(\tilde{G}_{j_{k-1}})$.
Teste für alle Wörter $w \in \Sigma^*$ mit $|w| \leq |s_k|$, ob $w \in L(\tilde{G}_{j_{k-1}})$. Mögen $\tilde{w}_0, ..., \tilde{w}_r$ diejenigen Wörter in lexikographischer Ordnung sein, die den Test erfolgreich bestanden haben. Berechne $\tilde{j}_k := M(\tilde{w}_0, ..., \tilde{w}_r)$.
(* Nach Konstruktion gilt jetzt $\{\tilde{w}_0, ..., \tilde{w}_r\} \subseteq L(G_{\tilde{j}_k})$, da M konsistent arbeitet. *)
Berechne die kanonische Nummer j_k von $L(\tilde{G}_{j_{k-1}}) \cup L(G_{\tilde{j}_k})$ und gib sie aus. Beginne Stufe $k+1$.

Nach Konstruktion arbeitet $\tilde{M}$ offenbar iterativ. Aus der Definition von $\tilde{M}$ folgt weiterhin, daß nach Verlassen der Stufe k stets $t_k^+ \subseteq L(\tilde{G}_{j_k})$. Da M streng-monoton arbeitet gilt für alle überhaupt jemals von $\tilde{M}$ produzierten Hypothesen nach Konstruktion auch $L(\tilde{G}_{j_k}) \subseteq L$. Schließlich liefert die Definition von $\tilde{M}$ auch unmittelbar $L(\tilde{G}_{j_k}) \subseteq L(\tilde{G}_{j_{k+m}})$ für alle $m \in \mathbb{N}$. Damit arbeitet $\tilde{M}$ streng-monoton. Möge jetzt j_k eine Hypothese mit $L = L(\tilde{G}_{j_k})$ sein. Dann tritt stets Fall 1 ein, so daß $\tilde{M}$ diese Hypothese in allen nachfolgenden Stufen wiederholt. Zum Nachweis der Konvergenz von $\tilde{M}$ auf t genügt es somit zu zeigen, daß $\tilde{M}$ irgendwann eine richtige Hypothese produziert.

Hilfssatz. $\tilde{M}$ konvergiert auf dem Text t.

Wir unterscheiden die folgenden zwei Fälle.

Fall A. L ist endlich.

Da L endlich ist, kann nach den obigen Beobachtungen von $\tilde{M}$ nur in endlich vielen Stufen k eine Inkonsistenz festgestellt werden. Jede festgestellte Inkonsistenz führt aber zu einem Hypothesenwechsel. Somit kann jedes Wort $w \in L$ maximal einen Hypothesenwechsel herbeiführen. Da $\tilde{M}$ weiterhin als Hypothesen nur Nummern j_k mit $L(\tilde{G}_{j_k}) \subseteq L$ ausgibt, muß irgendwann eine richtige Hypothese erreicht werden, da ein einmal im Text gesehenes Wort in jeder von einer weiteren Hypothese generierten Sprache enthalten ist. Damit ist der Hilfssatz für endliche Sprachen bewiesen.

Fall B. L ist unendlich.

Angenommen, $\tilde{M}$ wechselt unendlich oft die Hypothese. Da ein Hypothesenwechsel nur durch eine festgestellte Inkonsistenz erzwungen werden kann, gibt es somit unendlich viele *verschiedene* Wörter s, die $\tilde{M}$ zwingen, ihre aktuelle Hypothese zu verwerfen. Wir werden jetzt zeigen, daß dieses Scenario unmöglich ist. Möge dazu t^{ord} der lexikographisch geordnete Text von L sein. Da die IIM M insbesondere auch auf t^{ord} gegen eine für L korrekte Hypothese zu konvergieren hat, existiert ein $z \in \mathbb{N}$, so daß $M(t_z^{ord}) = j$ und $L = L(G_j)$. Da M nach Voraussetzung streng-monoton arbeitet muß weiterhin für alle Hypothesen $h = M(t_z^{ord}\sigma)$ die Bedingung $L = L(G_h)$ erüllt sein, wobei σ eine beliebige Folge von Wörtern aus L ist. Nun ist t ein Text für L. Somit existiert eine Zahl $x \in \mathbb{N}$ mit $Val\ t_z^{ord} \subseteq t_x^+$. Wie wir bereits gesehen haben, kann jedes Wort $w \in L$ die Maschine $\tilde{M}$ höchstens einmal zwingen, ihre Hypothese zu wechseln. Damit hat aber $\tilde{M}$ zumindest in der Stufe x eine Hypothese j_x mit $t_x^+ \subseteq L(\tilde{G}_{j_x})$ zu produzieren. Nach unserer Annahme wechselt $\tilde{M}$ jedoch unendlich oft die Hypothese. Also muß in irgendeiner Stufe $x+m$ ein Wort w mit $w \notin L(\tilde{G}_{j_{x+m-1}})$ als Eingabe vorliegen. Da $\tilde{M}$ streng-monoton arbeitet ist aber $L(\tilde{G}_{j_x}) \subseteq L(\tilde{G}_{j_{x+m-1}})$. Also muß weiterhin $w \notin L(\tilde{G}_{j_x})$ gelten. Nach Konstruktion berechnet die IIM $\tilde{M}$ jetzt eine Folge $\tilde{w}_0, ..., \tilde{w}_r$, die t_z^{ord} als Präfix enthält. Zur Bestimmung der auszugebenen Hypothese wird $M(\tilde{w}_0, ..., \tilde{w}_r) = M(t_z^{ord}\sigma) = h$ berechnet. Wie wir bereits gesehen haben, gilt somit $L = L(G_h)$. Da $L(\tilde{G}_{j_{x+m-1}}) \subseteq L$, ist folglich auch die kanonische Nummer von $L(\tilde{G}_{j_{x+m-1}}) \cup L(G_h)$ korrekt. Also gibt $\tilde{M}$ am Ende der Abarbeitung von Stufe $x+m$ eine korrekte Hypothese aus. Damit ist der Hilfssatz auch für unendliche Sprachen L gezeigt.

q.e.d.

Es sei noch angemerkt, daß es gelungen ist, eine Reihe weiterer Resultate bzgl. der Leistungsfähigkeit iterativ arbeitender Lernalgorithmen zu erzielen (cf., Lange und Zeugmann, 1992c).

4.3.3. Charakterisierung finiten Lernens

In diesem Abschnitt charakterisieren wir exaktes und klassentreues finites Lernen. Gemeinsam mit Satz 4.2 erhalten wir so eine vollständige Charakterisierung finiten Lernens bei Informant- bzw. Textbelehrung. Da $ESMON^d\!-\!TXT = SMON^d\!-\!TXT = FIN\!-\!TXT$ ist, charakterisieren wir gleichzeitig exakte und klassentreue dual strengmonotone Inferenz aus positiven Daten. Es sei noch angemerkt, daß Mukouchi (1992a) unabhängig von uns eine Charakterisierung exakten finiten Lernens erzielte. Der folgende Satz ist insofern strenger, als er gleichzeitig die Gleichheit von exaktem und klassentreuem finiten Lernen liefert.

Satz 4.39. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in FIN\!-\!TXT$ genau dann, wenn es eine rekursiv generierbare Familie $(T_j)_{j\in\mathbb{N}}$ endlicher, nichtleerer Mengen derart gibt, daß*

(1) *$T_j \subseteq L_j$ für alle $j \in \mathbb{N}$,*

(2) *für alle $k,\ z \in \mathbb{N}$: Wenn $T_z \subseteq L_k$, so $L_z = L_k$.*

Beweis: Notwendigkeit. Möge $\mathcal{L} \in FIN\!-\!TXT$ sein. Dann gibt es einen klassentreuen Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ und eine IIM M, so daß $\mathcal{L} \subseteq FIN\!-\!TXT(M)$ bzgl. $\mathcal{G}$. Wir zeigen jetzt, wie die Familie $(T_j)_{j\in\mathbb{N}}$ zu konstruieren ist. Die Definition erfolgt in zwei Schritten. Zunächst konstruieren wir eine Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ bzgl. des Hypothesenraumes $\mathcal{G}$. Danach geben wir eine Prozedur an, mit deren Hilfe die gewünschte Familie $(T_j)_{j\in\mathbb{N}}$ aus der Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ gewonnen wird.

Sei $k \in \mathbb{N}$ beliebig fixiert. Möge t^k der kanonische Text von $L(G_k)$ sein. Da die IIM M die Sprache $L(G_k)$ finit bei Belehrung durch t^k erlernt, existiert ein $x \in \mathbb{N}$, so daß $M(t^k_x) = m$ und $L(G_k) = L(G_m)$. Wir setzen $\hat{T}_k = Val\ t^k_x$. Damit ist die Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ vollständig definiert.

Die Prozedur zur Definition von $(T_j)_{j\in\mathbb{N}}$ arbeitet wie folgt. Bei Eingabe von $z \in \mathbb{N}$ wird das kleinste $j \in \mathbb{N}$ derart gesucht, daß $\hat{T}_j \subseteq L_z$. Wenn dieses j gefunden wurde, so setze $T_z = \hat{T}_j$.

Wir zeigen jetzt die oben annoncierten Eigenschaften. Da für jede Menge $\hat{T}_k$ ein Text t einer Sprache $L \in \mathcal{L}$ mit $\hat{T}_k \subseteq t^+$ existiert, muß die Suche nach dem ersten j in der gerade beschriebenen Prozedur stets terminieren. Damit sind alle $T_z \neq \emptyset$ und nach Konstruktion offensichtlich rekursiv generierbar und endlich. Aus der Definition der Familie $(T_j)_{j\in\mathbb{N}}$ ergibt sich direkt Eigenschaft (1). Zum Nachweis von (2) mögen $k, z \in \mathbb{N}$ mit $T_z \subseteq L_k$

sein. Nach Konstruktion existiert somit ein Index j mit $T_z = \hat{T}_j$. Aus der Definition der Familie $(\hat{T}_j)_{j\in\mathbb{N}}$ folgt weiterhin die Existenz eines eindeutig bestimmten Anfangsstückes t_x^j des kanonischen Textes t^j von $L(G_j)$, so daß $T_j = Val\ t_x^j$, $m = M(t_x^k)$ und $L(G_j) = L(G_m)$. Da $\hat{T}_j \subseteq L_k$, muß t_x^j auch Anfangsstück eines Textes für L_k sein. Da M die Sprache L_k auf jedem Text $t \in Text(L_k)$ finit lernt, erhalten wir somit $L_k = L(G_m)$. Aus der Definition von T_z folgt natürlich auch $T_z \subseteq L_z$. Die gleiche Argumentation liefert somit auch $L_z = L(G_m)$. Damit erhalten wir (2) aus der Transitivität der Gleichheitsrelation.

Hinlänglichkeit. Es genügt eine IIM M zu definieren, die $\mathcal{L}$ finit bzgl. $\mathcal{L}$ lernt. Mögen $L \in \mathcal{L}$ und $t \in Text(L)$ sein. Wir setzen:

$M(t) =$ „Lies t_0. Berechne T_0 und teste, ob $T_0 \subseteq t_0^+ \subseteq L_0$. Wenn ja, so gib 0 aus und stoppe. Andernfalls gib keine Hypothese aus und arbeite Stufe 1 ab.

Stufe x: Lies t_x. Generiere T_j für $j = 0, ..., x$ und teste, ob $T_j \subseteq t_x^+ \subseteq L_j$.

Wird das erste $j \leq x$ mit $T_j \subseteq t_x^+ \subseteq L_j$ gefunden, so gib j aus und stoppe.

Andernfalls gib keine Hypothese aus und arbeite Stufe $x+1$ ab.“

Da die Mengen T_j rekursiv generierbar und endlich sind, ist M eine IIM. Sei $k = \mu n[L = L_n]$. Dann muß ein $x \in \mathbb{N}$ mit $T_k \subseteq t_x^+$ existieren. Nach dem Lesen von t_x gibt M also spätestens eine Hypothese aus und stoppt. Folglich arbeitet M finit. Angenommen, M gibt eine Hypothese z mit $z \neq k$ aus und stoppt. Dann hat M die Bedingungung $T_z \subseteq t_y^+ \subseteq L_z$ für ein y verifiziert. Da t ein Text von L_k ist folgt somit $T_z \subseteq L_k$. Nach Eigenschaft (2) gilt somit $L_z = L_k$. Also gibt M stets eine korrekte Hypothese aus.

q.e.d.

Der letzte Satz erlaubt folgendes Corollar, das wir bereits im Beweis des Satzes 4.2 angewendet haben.

Corollar 4. 40. $EFIN{-}TXT = FIN{-}TXT$

Mit völlig analogen Methoden wie im Beweis des letzten Satzen zeigt man nachstehende Charakterisierung für $FIN-INF$. Aus dem Beweis erhält man zusätzlich auch das Analogon zu Corollar 4.40 für finites Lernen bei Informantbelehrung.

Satz 4.41. *Sei $\mathcal{L}$ eine indizierte Familie. Dann gilt: $\mathcal{L} \in FIN{-}INF$ genau dann, wenn es rekursiv generierbare Familien $(P_j)_{j\in\mathbb{N}}$ und $(N_j)_{j\in\mathbb{N}}$ von endlichen, nicht leeren Mengen derart gibt, daß*

(1) $\emptyset \neq P_j \subseteq L_j$ *and* $N_j \subseteq co{-}L_j$ *für alle* $j \in \mathbb{N}$,

(2) *für all* $k,\ j \in \mathbb{N}$*:* $P_k \subseteq L_j$ *and* $N_k \subseteq co{-}L_j$ *impliziert* $L_k = L_j$.

Corollar 4. 42. $EFIN{-}INF = FIN{-}INF$

Man kann mit ähnlichen Mittel, wie den hier vorgestellten, auch monotone und dual monotone Inferenz sowie Lernen unter einer begrenzten Anzahl von Hypothesenwechseln charakterisieren. Die Beweise sind allerdings deutlich komplizierter, so daß wir hier aus Platzgründen auf sie verzichten wollen. Wir verweisen statt dessen auf die Literatur (cf., Lange und Zeugmann, 1992c, 1993a) sowie Zeugmann, Lange und Kapur (1992). Es sei noch angemerkt, daß wir für alle hier eingeführten Lerntypen auch Charakterisierungen für die Fälle erzielt haben, in denen bei Informantbelehrung zu lernen ist. Um den Umfang dieser Arbeit in den gebotenen Grenzen zu halten, verzichten wir hier auf die Präsentation dieser Resulte und verweisen statt dessen auf Lange und Zeugmann (1993b).

4.4. Exaktes, klassentreues und klassenumfassendes Lernen

In diesem Abschnitt beschäftigen wir uns mit dem Problem, warum indizierte Familien bzgl. eines Hypothesenraumes $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ lernbar sein können, während sie sich bzgl. eines anderen Raumes $\hat{\mathcal{G}} = (\hat{G}_j)_{j \in \mathbb{N}}$ nicht inferieren lassen. Ein erster Ansatz zur Beantwortung dieser Frage findet sich bereits bei Gold (1967). Er konnte nämlich zeigen, daß die Existenz limes-rekursiver Compiler (cf., Definition 21 unten) von $\mathcal{G}$ nach $\hat{\mathcal{G}}$ hinreichend dafür ist, bzgl. $\mathcal{G}$ inferierbare indizierte Familien auch bzgl. $\hat{\mathcal{G}}$ lernen zu können. Mit unserem Satz 4.1 konnten wir zeigen, daß es zu jeder indizierten Familie $\mathcal{L}$, die überhaupt bzgl. eines Raumes $\mathcal{G}$ inferiert werden kann, einen limes-rekursiven Compiler von $\mathcal{G}$ nach $\mathcal{L}$ gibt. Die Situation ändert sich jedoch dramatisch, wenn zusätzlich Monotonieforderungen ins Spiel kommen. Das hat folgenden Grund. Die im Beweis von Satz 4.1 implizit enthaltene Konstrukion der limes-rekursiven Compiler erhält nicht die in Definition 17 eingeführten Monotoniebedingungen. Aber selbst wenn ein limes-rekursiver Compiler Monotoniebedingungen genügt, ist es eine nicht triviale Angelegenheit, eine beispielsweise bzgl. $\mathcal{G}$ streng-monoton lernende IIM so zu transformieren, daß sie auch bzgl. $\mathcal{L}$ streng-monoton lernt. Die Schwierigkeiten bei der Transformation sind auf die zu leistene Kombination zweier limes-rekursiver Prozesse zurückzuführen, die dabei die relevanten Monotoniebedingungen zu erhalten hat. Die Überwindung der kurz geschilderten Probleme gelang nun unter wesentlicher Ausnutzung der im letzten Abschnitt aufgestellten Charakterisierungssätze. Wir werden uns hier darauf beschränken, nur einen der erhaltenen Sätze vorzustellen, da er schon hinreichend guten Einblick in die Natur der

gewonnenen Erkenntnisse gestatten. Doch zunächst definieren wir limes-rekursive Compiler.

Definition 21. *Seien $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ und $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ zwei Hypothesenräume mit $L(\mathcal{G}) = L(\hat{\mathcal{G}})$.* ***Eine rekursive Funktion $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ wird limes-rekursiver Compiler von $\mathcal{G}$ nach $\hat{\mathcal{G}}$ genannt,*** *wenn für alle $j \in \mathbb{N}$ der Grenzwert $k = \lim_{x\to\infty} f(j,x)$ existiert und der Bedingung $L(G_j) = L(\hat{G}_k)$ genügt.*

Es sei noch angemerkt, daß alle vom Compiler berechneten Zahlen stets als als Hypothesen im Hypothesenraum $\hat{\mathcal{G}}$ interpretiert werden.

Möge jetzt $\mathcal{L}$ eine indizierte Familie, $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein für $\mathcal{L}$ klassentreuer Hypothesenraum und M eine bzgl. $\mathcal{G}$ streng-monoton lernende IIM sein. Wie der Beweis von Satz 14 zeigt, existiert nun stets ein limes-rekursiver Compiler f von $\mathcal{G}$ nach $\mathcal{L}$. Es ist im allgemeinen jedoch nicht möglich, aus M und f eine IIM $\hat{M}$ zu konstruieren, die $\mathcal{L}$ exakt streng-monoton lernt, da nach Satz 4.21 $ESMON{-}TXT \subset SMON{-}TXT$. Somit muß ein limes-rekursiver Compiler sicher zusätzlichen Forderungen genügen, wenn aus der klassentreuen streng-monotonen Lernbarkeit die exakte streng-monotone Inferenz ableitbar sein soll. Wir kombinieren deshalb den Begriff eines limes-rekursiven Compilers mit einer Monotonieforderung.

Definition 22. *Seien $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ und $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ zwei Hypothesenräume mit $L(\mathcal{G}) = L(\hat{\mathcal{G}})$.* ***Ein limes-rekursiver Compiler f von $\mathcal{G}$ nach $\hat{\mathcal{G}}$ heißt streng-monoton,*** *falls*

$$L(\hat{G}_{f(j,x)}) \subseteq L(\hat{G}_{f(j,x+1)}) \text{ für alle } j,x \in \mathbb{N} \text{ gilt.}$$

Nun können wir die gewünschte Charakterisierung angeben, die exaktes und klasssentreues streng-monotones Lernen vergleicht.

Satz 4.43. *Seien $\mathcal{L}$ eine indizierte Familie und $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ ein Hypothesenraum, so daß $\mathcal{L} \in SMON{-}TXT$ bzgl. $\mathcal{G}$ ist. Dann gilt:*

$\mathcal{L} \in ESMON{-}TXT$ genau dann, wenn es einen streng-monotonen limes-rekursiven Compiler von $\mathcal{G}$ nach $\mathcal{L}$ gibt.

Beweis: Notwendigkeit. Sei $\mathcal{L} \in ESMON{-}TXT$. Dann existiert eine IIM M, so daß $\mathcal{L} \subseteq ESMON{-}TXT(M)$ bzgl. $\mathcal{L}$. Wir definieren den gewünschten limes-rekursiven Compiler von $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ nach $\mathcal{L}$ wie folgt. Seien $j,x \in \mathbb{N}$ und t^j der kanonische Text von $L(G_j)$. Wir setzen:

$f(j,x) =$ „Berechne die Folge $(M(t^j_z))_{z\in\mathbb{N}}$ bis zur Länge $x+1$. Möge j_y das letzte Glied

dieser Folge sein. Setze $f(j,x) = j_y$.“

Wir haben noch zu zeigen, daß der definierte Compiler f das Verlangte leistet. Nach Konstruktion ist offenbar $f \in \mathcal{R}^2$. Möge $j \in \mathbb{N}$ beliebig fixiert sein. Da $\mathcal{G}$ ein klassentreuer Hypothesenraum ist, gibt es einen Index m mit $L(G_j) = L_m$. Folglich lernt die IIM M die Sprache $L(G_j)$ insbesondere bei Belehrung durch t^j. Somit konvergiert die Folge $(M(t^j_z))_{z\in\mathbb{N}}$ im Limes gegen eine Zahl k mit $L_k = L(G_j)$. Also gilt nach Konstruktion $k = \lim_{x\to\infty} f(j,x)$ und $L_k = L(G_j)$. Damit ist f limes-rekursiver Compiler von $\mathcal{G}$ nach $\mathcal{L}$.

Mögen jetzt $f(j,x) = j_y$ und $f(j,x+1) = j_n$ sein. Da M streng-monoton arbeitet, erhalten wir $L_{j_y} \subseteq L_{j_n}$. Somit ist f streng-monotoner limes-rekursiver Compiler.

Hinlänglichkeit. Seien $\mathcal{L} \subseteq SMON-TXT(\hat{M})$ bzgl. $\mathcal{G}$ und f ein streng-monotoner limes-rekursiver Compiler von $\mathcal{G}$ nach $\mathcal{L}$. Wir haben eine IIM M derart zu definieren, daß $\mathcal{L} \subseteq ESMON-TXT(M)$. Die entscheidene Schwierigkeit des Beweises besteht darin, zwei limes-rekursive Prozesse so zu verknüpfen, daß die resultierende IIM M nicht nur $\mathcal{L}$ lernt, sondern auch streng-monoton arbeitet. Möge t irgendein Text einer Sprache $L \in \mathcal{L}$ sein. Weiterhin sei $j_x = \hat{M}(t_x)$ und $j_{x+r} = \hat{M}(t_{x+r})$. Dann gilt zwar $L(G_{j_x}) \subseteq L(G_{j_{x+r}})$, aber es ist zunächst völlig unklar, in welcher Relation die vom limes-rekursiven Compiler f berechneten Hypothesen $L_{f(j_x,z)}$ und $L_{f(j_{x+r},m)}$ stehen, bzw. wie z und m zu wählen sind, damit $L_{f(j_x,z)} \subseteq L_{f(j_{x+r},m)}$ gilt. Eine Idee zur Überwindung dieses Problems bietet Satz 4.37 an, nämlich geeignet rekursiv generierbare endliche Mengen zu definieren, mit deren Hilfe das Inklusionsproblem rekursiv beherrscht werden kann. Wir beweisen deshalb zunächst folgende Lemmata.

***Lemma* 1**. *Zum Hypothesenraum $\mathcal{G}$ existiert eine rekursiv generierbare Familie endlicher Mengen $(T^y_j)_{j,y\in\mathbb{N}}$, so daß folgende Eigenschaften erfüllt sind:*

(1) *für alle $L \in \mathcal{L}$ existiert ein j mit $L = L(G_j)$ und $T^y_j \neq \emptyset$ für fast alle $y \in \mathbb{N}$,*

(2) *für alle $j,y \in \mathbb{N}$ gilt: $T^y_j \subseteq L(G_j)$ und $T^y_j \neq \emptyset$ impliziert $T^{y+1}_j = T^y_j$,*

(3) *für alle $j,y,z \in \mathbb{N}$ gilt: $\emptyset \neq T^y_j \subseteq L_z$ bzw. $\emptyset \neq T^y_j \subseteq L(G_z)$ impliziert $L(G_j) \subseteq L_z$ bzw. $L(G_j) \subseteq L(G_z)$.*

Wir definieren die gewünschten Mengen T^y_j wie folgt. Seien $j,y \in \mathbb{N}$; dann setzen wir:

$$T^y_j = \begin{cases} Val\ t^j_z, & \text{falls} \quad z = min\{x \mid x \leq y,\ \hat{M}(t^j_x) = j\}, \\ \emptyset, & \text{sonst.} \end{cases}$$

wobei $\hat{M}$ die IIM ist, die $\mathcal{L}$ bzgl. $\mathcal{G}$ streng-monoton lernt. Offenbar sind die Mengen T_j^y rekursiv generierbar und endlich. Der Nachweis von (1) bis (3) erfolgt mutatis mutandis wie im Beweis von Satz 4.37.

q.e.d. (Lemma 1)

Lemma 2. *Möge $(T_j^y)_{j,y\in\mathbb{N}}$ die Familie endlicher rekursiv generierbarer Mengen aus Lemma 1 sein. Dann gibt es eine IIM $\tilde{M}$, so daß folgende Eigenschaften erfüllt sind:*

(1) $\mathcal{L} \subseteq SMON\text{-}TXT(\tilde{M})$,

(2) *für alle $L \in \mathcal{L}$ und alle $t \in Text(L)$ gilt: $z = \lim_{x\to\infty} \tilde{M}(t_x)$ impliziert $T_z^y \neq \emptyset$ für fast alle $y \in \mathbb{N}$.*

Wir definieren die gewünschte IIM $\tilde{M}$ ähnlich wie im Beweis der Hinlänglichkeit von Satz 4.37. Mögen $L \in \mathcal{L}$, $t \in Text(L)$ und $x \in \mathbb{N}$ sein. Weiterhin sei $(T_j^y)_{j,y\in\mathbb{N}}$ die rekursiv generierbare Familie aus Lemma 1. Wir setzen:

$\tilde{M}(t_x) =$ „Generiere für $j = 1, ..., x$ die Mengen T_j^x und teste für alle nichtleeren T_j^x, ob $T_j^x \subseteq t_x^+ \subseteq L(G_j)$. Wird wenigstens ein j gefunden, das den Test erfüllt, so gib das kleinste derartige j aus und lies anschließend die nächste Eingabe.

Andernfalls gib keine Hypothese aus und lies die nächste Eingabe."

Wir zeigen jetzt, daß $\tilde{M}$ das Verlangte leistet.

Hilfssatz. $\tilde{M}$ lernt $\mathcal{L}$.

Sei $k = \mu z[L = L(G_z)$ und $T_z^y \neq \emptyset$ für fast alle $y]$. Nach Aussage (1) von Lemma 1 ist die Existenz dieses k gesichert. Weiterhin liefert Aussage (2) von Lemma 1, daß es ein y_0 mit folgender Eigenschaft gibt:

$$T_k^0 = ... = T_k^{y_0} = \emptyset \text{ und } \emptyset \neq T_k^{y_0+1} = T_k^{y_0+r} \text{ für alle } r \in \mathbb{N}.$$

Die letzte Eigenschaft gilt natürlich auch für alle j, für die überhaupt ein y mit $T_j^y \neq \emptyset$ existiert. Außerdem ist somit klar, daß eine einmal verworfene Hypothese in keinem nachfolgenden Lernschritt ausgegeben wird. Der Rest kann somit völlig analog wie im Beweis des Hilfssatzes 1 von Satz 4.37 gezeigt werden.

q.e.d. (Hilfssatz)

In völliger Analogie zum Beweis von Hilfssatz 2 in der Demonstration von Satz 4.37 zeigt man weiterhin, daß $\tilde{M}$ streng-monoton arbeitet.

Eigenschaft (2) erhält man direkt aus der Definition von $\tilde{M}$ sowie der im Beweis des Hilfsatzes gemachten Beobachtung.

q.e.d. (Lemma 2)

Wir kommen jetzt zur Definition der gewünschten IIM M. Seien $L \in \mathcal{L}$, $t \in \mathit{Text}(L)$ und $x \in \mathbb{N}$. Mögen $(T_j^y)_{j,y \in \mathbb{N}}$ die rekursiv generierbare Familie aus Lemma 1 und $\tilde{M}$ die IIM aus Lemma 2 sein. Wir setzen:

$M(t_x) =$ „Wenn $x = 0$ oder M bis zum Lesen von t_{x-1} noch keine Hypothese produziert hat, so arbeite (A) ab. Anderfalls führe (B) aus.

(A) Simuliere $\tilde{M}$ bis zum Lesen von t_x. Gibt $\tilde{M}$ keine Hypothese aus, so lies die nächste Eingabe und gib nichts aus. Andernfalls sei $j = \tilde{M}(t_x)$. Setze $FLAG(x) = j$, gib $f(j,x)$ aus und lies die nächste Eingabe.
(* Nach der Definition von $\tilde{M}$ wissen wir, daß $T_j^x \neq \emptyset$. *)

(B) Simuliere $\tilde{M}$ bis zum Lesen von t_x. Gibt $\tilde{M}$ nach dem Lesen von t_x keine Hypothese aus, so setze $FLAG(x) = FLAG(x-1)$, gib $f(FLAG(x),x)$ aus und lies die nächste Eingabe.
Andernfalls sei $k = \tilde{M}(t_x)$.
(* Nach der Definition von $\tilde{M}$ ist $T_k^x \neq \emptyset$. *)
Ist $k = FLAG(x-1)$, so setze $FLAG(x) = FLAG(x-1)$, gib $f(FLAG(x),x)$ aus und lies die nächste Eingabe.
Wenn $k \neq FLAG(x-1)$, so teste, ob $T^x_{FLAG(x-1)} \subseteq L_{f(k,x)}$.
Gilt $T^x_{FLAG(x-1)} \not\subseteq L_{f(k,x)}$, so setze $FLAG(x) = FLAG(x-1)$, gib $f(FLAG(x),x)$ aus und lies die nächste Eingabe.
Ist hingegen $T^x_{FLAG(x-1)} \subseteq L_{f(k,x)}$, so setze $FLAG(x) = k$, gib $f(FLAG(x),x)$ aus und lies die nächste Eingabe.“

Wir haben noch zu beweisen, daß M die Familie $\mathcal{L}$ streng-monoton bzgl. $\mathcal{L}$ lernt.

Beh. 1. M arbeitet streng-monoton.

Da f ein streng-monotoner limes-rekursiver Compiler von $\mathcal{G}$ nach $\mathcal{L}$ ist, gilt für alle $L \in \mathcal{L}$ zunächst folgendes. Wenn $L(G_j) = L$, so ist $L_{f(j,x)} \subseteq L$ für alle $x \in \mathbb{N}$. Das sieht man wie folgt. Sei $k = lim_{x \to \infty} f(j,x)$. Nun ist f insbesondere limes-rekursiver Compiler. Somit erhalten wir $L(G_j) = L_k = L$. Da f auch streng-monoton arbeitet, gilt überdies

$$L_{f(j,x)} \subseteq L_{f(j,x+1)} \subseteq ... \subseteq L_{lim_{x \to \infty} f(j,x)} = L_k = L = L(G_j) \qquad (E)$$

Aus dieser Beobachtung erhalten wir unmittelbar, daß f streng-monoton arbeitet, sofern $FLAG(x) = FLAG(x-1)$.

Möge jetzt $FLAG(x) \neq FLAG(x-1)$ sein. Wenn M ihre erste Hypothese ausgibt, so gilt $T_j^x \neq \emptyset$ nach Definition von M. Nach Aussage (2) von Lemma 1 ist somit auch $T_j^{x+r} \neq \emptyset$ für alle $r \in \mathbb{N}$. Wechselt M nach dem Lesen von t_x die Hypothese, etwa von j zu k, so ist nach dem Kommentar in der Definition von M auch $T_k^x \neq \emptyset$ gesichert. Aussage (2) von Lemma 1 liefert dann $T_k^{x+r} \neq \emptyset$ für alle $r \in \mathbb{N}$. Somit ist stets $T_{FLAG(x-1)}^x \neq \emptyset$ gesichert. Also haben wir insgesamt nach der Definition von M die folgende Konstellation: $k = \tilde{M}(t_x)$, $k \neq FLAG(x-1)$ und $\emptyset \neq T_{FLAG(x-1)}^x \subseteq L_{f(k,x)} = L_{f(FLAG(x),x)}$.

Nach Konstruktion, der Definition von f und der oben gemachten Beobachtung (E) ist zunächst

$$L_{f(FLAG(x-1),x-1)} \subseteq L(G_{FLAG(x-1)}).$$

Außerdem sind die Voraussetzungen der Eigenschaft (3) von Lemma 1 erfüllt. Folglich erhalten wir:

$$T_{FLAG(x-1)}^x \subseteq L_{f(k,x)} \text{ impliziert } L(G_{FLAG(x-1)}) \subseteq L_{f(k,x)}.$$

Insgesamt ergibt sich somit:

$$L_{f(FLAG(x-1),x-1)} \subseteq L_{f(k,x)} = L_{f(FLAG(x),x)}$$

Damit arbeitet M streng-monoton.

Beh. 2. M lernt L bei Belehrung durch t.

Nach Voraussetzung wissen wir zunächst, daß $z = lim_{x\to\infty}\tilde{M}(t_x)$ existiert und weiterhin $L = L(G_z)$ gilt. Aussage (2) von Lemma 2 sichert außerdem die Existenz eines y_0, so daß $T_z^y \neq \emptyset$ für alle $y > y_0$. Da f insbesondere limes-rekursiver Compiler von $\mathcal{G}$ nach $\mathcal{L}$ ist, erhalten wir noch folgendes: für alle j mit $L = L(G_j)$ existiert $n = lim_{x\to\infty} f(j,x)$ und es gilt $L_n = L(G_j) = L$.

Nach Konstruktion gibt M somit irgendwann eine erste Hypothese $f(j,x)$ aus. Ist $j = z$, so sind alle weiteren Hypothesen von der Form $f(j, x+r)$, $r \in \mathbb{N}$. In diesem Falle konvergiert M also gegen eine korrekte Hypothese. Möge nun $j \neq z$ sein. Dann hat $\tilde{M}$ noch nicht auf t konvergiert. Es genügt somit zu zeigen, daß für hinreichend große x nur noch Hypothesen der Form $f(z,x)$ von M ausgegeben werden. Ist $x > y_0$, so ist $T_z^x \neq \emptyset$. Für hinreichend große x ist überdies $L = L_{f(z,x)}$. Da, wie wir oben gesehen haben, $L(G_{FLAG(x-1)}) \subseteq L$ und nach Aussage (2) von Lemma 1 auch $T_{FLAG(x-1)}^x \subseteq L$, muß M für ein hinreichend großes x diese Bedingungen verifizieren und $FLAG(x) = z$ setzen. Damit ist dann aber kein weiterer Hypothesewechsel mehr möglich, da dazu $\tilde{M}$ nochmals ihre Hypothese wechseln müßte. Also konvergiert M gegen eine korrekte Hypothese für L.

q.e.d.

Es ist weiterhin auch für schwach-monotones Lernen gelungen, charakteristische Bedingungen für die Äquivalenz von klassentreuer und exakter Inferenz anzugeben. Aus Platzgründen verzichten wir hier auf die Darstellung und verweisen statt dessen auf Lange und Zeugmann (1993c).

Wir wenden uns nun einer inhaltlich sehr wichtigen Frage zu. Wie wir in Sektion 4.1 gesehen haben, ist klassenumfassendes Lernen immer leistungsfähiger als klassentreue Inferenz, die ihrerseits wieder fast immer über größere Potenzen verfügt als exakte Erkennung. Insbesondere zeigen die Sätze 4.20, 4.23 und 4.21 nachdrücklich, daß es im allgemeinen nicht genügt, nur die Beschreibungen und die Aufzählung einer indizierten Familie zu ändern, um Lernalgorithmen höchster Leistungsfähigkeit zu erhalten. Es ist vielmehr sehr bedeutsam, die Hypothesenräume geeignet anzureichern. Somit entsteht natürlich die Frage, welche Art von Sprachen einem Hypothesenraum hinzuzufügen sind. Besonders interessant ist unseres Erachtens das Problem, ob man Sprachen zum Hypothesenraum hinzunehmen sollte, die ihrerseits selbst nicht erlernbar sind. Wie die folgenden Sätze zeigen, hängt die Antwort wesentlich von den involvierten Monotoniebedingungen ab. Analysiert man den Beweis von Satz 4.23, so erhält man das folgende, auf den ersten Blick überraschende, Resultat.

Satz 4.44. *Für alle indizierten Familien $\mathcal{L}$ gilt:*

Wenn $\mathcal{L} \in KWMON^d\text{–}TXT$, so existert ein die Familie $\mathcal{L}$ umfassender Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ derart, daß $(L(G_j))_{j\in\mathbb{N}} \in EWMON^d\text{–}TXT$.

Eine Möglichkeit zur Konstruktion dieses Raumes $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ haben wir im Beweis von Satz 4.23 vorgestellt, i.e. man kann stets den Abschluß von $\mathcal{L}$ bzgl. endlicher Durchschnitte bilden. Diese Konstruktion ist also sogar völlig unabhängig von den jeweiligen IIMs, die $\mathcal{L}$ überhaupt lernen können.

Betrachten wir dual streng-monotone Inferenz, so ändert sich die Situation vollständig. In diesem Fall ist man nämlich gezwungen, Grammatiken zum Hypothesenraum hinzuzufügen, die ihrerseits nicht dual streng-monoton erlernbare Sprachen generieren.

Satz 4.45. *Sei $\mathcal{L}$ eine indizierte Familie mit $\mathcal{L} \in KSMON^d\text{–}TXT \setminus SMON^d\text{–}TXT$. Dann gibt es keinen Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ derart, daß $(L(G_j))_{j\in\mathbb{N}} \in ESMON^d\text{–}TXT$.*

Beweis: Wir führen den Beweis indirekt. Angenommen, es gibt einen Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$, so daß $(L(G_j))_{j\in\mathbb{N}} \in ESMON^d\text{–}TXT$ bzgl. $(L(G_j))_{j\in\mathbb{N}}$. Da nach Voraussetzung $\mathcal{L} \in KSMON^d\text{–}TXT \setminus SMON^d\text{–}TXT$, erhalten wir $\mathcal{L} \subset \{L(G_j) \mid j \in \mathbb{N}\}$.

Nach Satz 4.21 ist $FIN{-}TXT = SMON^d{-}TXT$, also folgt aus der Annahme sofort $(L(G_j))_{j\in\mathbb{N}} \in FIN{-}TXT$. Nach Corollar 4.40 ist $FIN{-}TXT = EFIN{-}TXT$. Somit gibt es eine IIM M, die $(L(G_j))_{j\in\mathbb{N}}$ finit bzgl. $(L(G_j))_{j\in\mathbb{N}}$ lernt. Da $\mathcal{L} \subset \{L(G_j) \mid j \in \mathbb{N}\}$, lernt M deshalb auch $\mathcal{L}$ finit. Folglich wäre, wieder nach Satz 4.21, $\mathcal{L} \in SMON^d{-}TXT$, im Widerspruch zur Vorraussetzung.

q.e.d.

Der letzte Satz dieses Abschnittes unterscheidet sich zumindest konzeptionell von den beiden vorangegangenen Sätzen. Während man im Falle dual streng-monotonen Lernens überhaupt keinen klassenumfassenden Hypothesenraum konstruieren kann, der selbst exakt erlernbar ist, gab es für dual schwach-monotone Inferenz sogar ein universelles und einfaches Verfahren zur Vervollständigung des Hypothesenraumes. Wie verhält es sich nun für schwach-monotones Lernen. Unsere Antwort hat zwei besondere Aspekte. Einerseits ist es gelungen, ein zu Satz 4.44 analoges Resultat zu zeigen. Andererseits ging dabei die zumindest die Einfachheit verloren. Wenn eine indizierte Familie $\mathcal{L}$ bzgl. eines Hypothesenraumes $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ schwach-monoton erlernbar ist, so gibt es jetzt nur noch ein in $\mathcal{L}$, $\hat{\mathcal{G}}$ und der $\mathcal{L}$ inferierenden IIM M uniformes Verfahren, das aus diesen drei Eingaben eine rekusive Aufzählung eines $\mathcal{L}$ umfassenden Hypothesenraumes $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ derart generiert, daß $L(G_j)_{j\in\mathbb{N})}$ exakt schwach-monoton erlernbar ist.

Satz 4.46. *Für alle indizierten Familien $\mathcal{L}$ gilt:*

Wenn $\mathcal{L} \in KWMON{-}TXT$, so existert ein die Familie $\mathcal{L}$ umfassender Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ derart, daß $(L(\hat{G}_j))_{j\in\mathbb{N}} \in EWMON{-}TXT$.

Beweis: Möge $\mathcal{L} \in KWMON-TXT$ sein. Dann existieren ein Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ und eine IIM M, so daß $\mathcal{L} \subseteq KWMON{-}TXT(M)$ bzgl. $\mathcal{G}$. Wir wenden jetzt Satz 4.36 an und erhalten einen Hypothesenraum $\hat{\mathcal{G}} = (\hat{G}_j)_{j\in\mathbb{N}}$ sowie eine Familie endlicher, nichtleerer und rekursiv generierbarer Mengen $(\hat{T}_j)_{j\in\mathbb{N}}$. Bezüglich dieses Hypothesenraumes $\hat{\mathcal{G}}$ definiert man wie im Beweis von Satz 4.34 eine IIM $\hat{M}$. Diese Maschine lernt gerade $(L(\hat{G}_j))_{j\in\mathbb{N}}$ bzgl. des Hypothesenraumes $\hat{\mathcal{G}}$.

q.e.d.

4.5. Ausblick

Wir diskutieren jetzt die im 4. Kapitel insgesamt erzielten Resultate hinsichtlich ihrer Bedeutung für die Inferenz formaler Sprachen. Mit den in Abschnitt 2.3 eingeführten Formalisierungen des Lernens durch Generalisieren bzw. Spezialisieren ist es unseres Erachtens gelungen, wesentliche Aspekte derartiger Inferenzmodi einer theoreti-

schen Untersuchung zugänglich zu machen. Gleichzeitig gestatteten die vorgenommenen Begriffsbildungen ein gründliches Studium der Frage, inwieweit Aspekte nichtmonotonen Denkens im Lernprozeß zu berücksichtigen sind. Dabei zeigen insbesondere Satz 4.23 und Corollar 4.17, daß sich Lernalgorithmen höchster Leistungsfähig nur entwerfen lassen, wenn konsequent auf Monotonieforderungen verzichtet wird. Obwohl sich dieses Bild etwas relativiert, wenn bei Informantbelehrung gelernt wird, bleibt die Bedeutung der Modellierung nichtmonotoner Aspekte des Denkens erhalten. Aus erkenntnistheoretischer Sicht verdienen die Sätze zur Leistungsfähigkeit konservativer Lernverfahren bei Textbelehrung besondere Beachtung. Einerseits zeigen Satz 4.13 und Corollar 4.15, daß sich die durchaus wünschenswerte Eigenschaft, semantisch finit zu lernen, uniform durch konservativ arbeitende Algorithmen realisieren läßt. Damit haben wir gleichzeitig eine uniforme Möglichkeit aufgezeigt, wie Lernalgorithmen ihre Hypothesen ggf. falsifizieren können. Andererseits zeigt Satz 4.23 in Verbindung mit Satz 4.22 die Grenzen konservativen Lernverhaltens deutlich auf. Mit diesen Sätzen konnten wir außerdem einen wichtigen Unterschied zum Lernen rekursiver Funktionen aufzeigen, wo insbesondere stets semantische Finitheit beim Lernen im Limes erreichbar ist. Beim Lernen aus positiven Daten scheint es hingegen so, daß eine IIM auch mal grundlos ihre Hypothese verwerfen muß. Diese Einschätzung spiegelt aber nur eine lokale Sichtweise des Lernprozesses wider. Gründe zum Verlassen einer bereits richtigen Hypothese leiten sich jetzt nämlich aus der globalen Forderung, alle Sprachen einer indizierten Familie lernen zu müssen, her. Solange nun die Topologie einer indizierten Familie rekursiv beherrschbar ist, kann konservativ gelernt werden. Reduziert sich die algorithmische Beherrschbarkeit der topologischen Struktur jedoch auf rekursive Aufzählbarkeit charakteristischer Eigenschaften, so ist konservatives Lernverhalten nicht mehr realisierbar. Die theoretische Begründung der gerade gemachten Aussagen ergibt sich unmittelbar aus den Sätzen 4.33 und 4.34.

Insbesondere ist mit Satz 4.34 auch die Lösung eines seit mehr als einer Dekade offenen Problems gelungen. Mit den dabei entwickelten Techniken gelang es überdies, fast alle Typen des monotonen bzw. dual monotonen Lernens zu charakterisieren. Diese Sätze erlauben natürlich auch vertiefte Einsichten in die Problematik, welche indizierten Familien unter bestimmten Monotonieforderungen inferierbar sind. Wir diskutieren hier nur den Fall klassentreuen streng-monotonen Lernens etwas näher. Für exaktes streng-monotones Lernen gelten die folgenden Bemerkungen mutatis mutandis. Betrachtet man Eigenschaft (3) in Satz 4.37 genauer, so stellt man folgendes fest. Wenn eine indizierte Familie $\mathcal{L}$ klassentreu streng-monoton gelernt werden kann, so gibt es eine rekursive Aufzählung von Grammatiken für alle in $\mathcal{L}$ enthaltenen Sprachen bezüglich derer das Teilsprachenproblem uniform entscheidbar ist. Die Aufzählung wird gerade durch den im Beweis von Satz

4.37 konstruierten Hypothesenraum geliefert. Um jetzt für beliebige $j, k \in \mathbb{N}$ entscheiden zu können, ob $L(G_j) \subseteq L(G_k)$ ist, genügt es, Eigenschaft (2) und (3) anzuwenden. Man generiert rekursiv T_j und testet, ob $T_j \subseteq L(G_k)$. Dieser Test ist effektiv ausführbar, da das Wortproblem in $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$ uniform rekursiv ist. Gilt nun die Inklusion $T_j \subseteq L(G_k)$, so folgt aus Eigenschaft (3) von Satz 4.37, daß $L(G_j) \subseteq L(G_k)$. Andererseits ergibt sich aus Eigenschaft (2) unmittelbar, daß auch die Inklusion $T_j \subseteq L(G_k)$ erfüllt sein muß, sofern $L(G_j) \subseteq L(G_k)$. Die Entscheidbarkeit des Teilsprachenproblems impliziert darüber hinaus sofort die Rekursivität des Äquivalenzproblems im Hypothesenraum $\mathcal{G} = (G_j)_{j \in \mathbb{N}}$. Somit kann der Hypothesenraum sogar stets so gewählt werden, daß er für jede Sprache $L \in \mathcal{L}$ genau eine L generierende Grammatik enthält. Während die im Beweis von Satz 4.37 angegebene IIM nicht notwendig semantisch finit arbeitet, erlaubt die letzte Beobachtung die Transformation von M in eine semantisch finite Maschine. Dazu ist die im Beweis von Satz 4.37 definierte IIM lediglich so zu modifizieren, daß sie über dem Hypothesenraum arbeitet, der jede Sprache aus $\mathcal{L}$ genau einmal aufzählt.

Wir wollen jetzt noch ein Beispiel dafür angeben, wie mit Hilfe der erzielten Sätze die Frage der algorithmischen Lösbarkeit von Lernproblemen beantwortet werden kann. Dazu betrachten wir die indizierte Familie aller Patternsprachen. Angluin (1980a) definierte Pattern und Patternsprachen wie folgt. Sei $\Sigma = \{a, b...\}$ ein beliebig fixiertes Alphabet und $X = \{x_1, x_2, ...\}$ eine zu Σ disjunkte Menge von Variablen. Pattern sind nichtleere Wörter über $\Sigma \cup X$, z.B. ax_1b, $x_1x_1bx_2$. Sei p ein Pattern. Die von p generierte Sprache $L(p)$ ist die Menge aller Wörter $w \in \Sigma^*$, die man aus p durch Substitution der Variablen durch nichtleere Wörter über Σ erhält.

Jantke (1991b) zeigte, daß die Familie aller Patternsprachen exakt streng-monoton bei Textbelehrung erlernbar ist, sofern für beliebige Pattern p und q entschieden werden kann, ob $L(p) \subseteq L(q)$. Es blieb in Jantke (1991b) aber offen, ob auch die Umkehrung gilt. Wie wir oben gesehen haben, impliziert insbesondere die exakte streng-monotone Lernbarkeit einer indizierten Familie die Entscheidbarkeit des Inklusionsproblems. Also ergäbe sich aus der exakten streng-monotonen Inferierbarkeit der Familie aller Patternsprachen die Entscheidbarkeit des Inklusionsproblems für Pattersprachen. Jiang, A. Salomaa, K. Salomaa und Yu (1993) zeigten nun, daß das Inklusionsproblem für Patternsprachen unentscheidbar ist. Folglich kann die Familie aller Patternspachen nicht exakt streng-monoton gelernt werden. Interessanterweise ist aber die größere Familie aller endlichen Durchschnitte von Pattersprachen klassentreu streng-monoton erlernbar (cf., Lange und Zeugmann, 1992a).

Die Charakterisierungen in Abschnitt 4.3 gestatten darüber hinaus eine unifizierende

Sicht monotoner bzw. dual monotoner Lernalgorithmen. Die in den Beweisen der Hinlänglichkeit definierten Inferenzmaschinen sind jeweils universell einsetzbar zur Lösung der gerade charakterisierten Lernprobleme. Beim Entwurf von Inferenzprozeduren kann man sich somit auf die rekursiv generierbaren Familien $(\hat{T}_j)_{j\in\mathbb{N}}$ endlicher nichtleerer Mengen konzentrieren. Beim Lernen aus positiven und negativen Daten fällt die Vereinheitlichung noch gravierender aus. Wie unsere Resultate in Lange und Zeugmann (1993b) zeigen, existiert ein Lernalgorithmus, der jedes Inferenzproblem unter Monotoniebedingungen realisiert.

Schließlich erlauben es die vorgestellten Charakterisierungssätze auch, weitere interessante Beziehungen zu anderen Bereichen des induktiven Lernens herzustellen. Freivalds, Kinber und Wiehagen (1993) definierten und studierten erstmals Lernen aus guten Beispielen. Dabei erhält eine IIM nicht mehr sukzessive den Graphen einer Funktion, sondern nur eine *endliche* Menge von Paaren Argument-Wert der zu inferierenden Funktion f, die insbesondere die guten Beispiele für f umfaßt. Eine IIM lernt f finit aus guten Beispielen, falls sie eine für f korrekte Hypothese produziert und dann stoppt. Überraschenderweise ist finite Inferenz aus guten Beispielen genauso leistungsfähig wie verhaltenskorrektes Lernen. Den gleichen Ansatz kann man natürlich auch im Gebiet des Lernens formaler Sprachen untersuchen. Erste Resultate hierzu finden sich in Lange und Wiehagen (1991). Die in den Charakterisierungen auftretenen endlichen Tell-tale Mengen können zumindest im streng-monotonen Fall unmittelbar als gute Beispiele eingesetzt werden. Eine aus diesen Beispielen bzw. Übermengen davon lernende IIM sucht im Hypothesenraum einfach die bzgl. Inklusion kleinste Sprache, die alle Elemente der Beispielmenge enthält.

Unserer Untersuchungen haben weiterhin ergeben, daß der sorgfältigen Wahl des Hypothesenraumes eine besondere Bedeutung zukommt, sofern unter Monotonieforderungen gelernt werden soll. Wie die Sätze 4.19, 4.20 und 4.23 nachdrücklich zeigen, kann die Lernfähigkeit monotoner bzw. dual monotoner Inferenzprozeduren um mehrere Größenordnungen erweitert bzw. reduziert werden, je nachdem, ob geeignete Hypothesenräume gewählt werden oder nicht. Es ist weiterhin gelungen, hinreichende und notwendige Bedingungen dafür anzugeben, unter welchen Bedingungen Hypothesenräume äquivalent sind.

Wir wollen die inhaltliche Bedeutung derartiger Resultate am Beispiel von Satz 4.43 erklären. Wenn man aus bestimmten Gründen daran interessiert ist, eine indizierte Familie $\mathcal{L}$ exakt streng-monoton zu lernen, so bietet Satz 4.43 folgende Möglichkeit des Entwurfs eines adäquaten Lernalgorithmus an. Man kann sich zunächst einen klassentreuen Hypothesenraum $\mathcal{G} = (G_j)_{j\in\mathbb{N}}$ definieren, der mit lernfreundlichen Eigenschaften

ausgestattet ist und bzgl. dieses Raumes eine IIM konstruieren. Zur Erreichung des ursprünglichen Lernziels genügt nun das Design eines streng-monotonen limes-rekursiven Compilers von $\mathcal{G}$ nach $\mathcal{L}$. Diese Aufgabe ist einfacher, da es jetzt ausreicht, für jede Sprache nur *einen* Text zu betrachten, während beim Entwurf eines Lernverfahrens potentiell *alle* Texte ins Kalkül zu ziehen sind.

Schließlich liefern die Sätze 4.44, 4.45 und 4.46 erste Ideen für das Design von Hypothesenräumen, die in gewisser Weise über lernfreundliche Eigenschaften verfügen. Weitere Forschungen in diese Richtung sind nötig, um wenigstens Heuristiken zu entwickeln, deren Einsatz den Entwurf von adäquaten Hypothesenräumen gestattet. Weitere Forschungen sollten sich außerdem mit der Frage beschäftigen, ob die Lernfähigkeit der in dieser Arbeit definierten Inferenztypen noch gesteigert werden kann, falls auf die uniforme Entscheidbarkeit des Wortproblems verzichtet wird. Wir vermuten interessante Resultate in dieser Richtung.

Abschließend diskutieren wir noch eine Frage, deren Beantwortung ebenfalls interessante Resultate erwarten läßt. Inhaltlich knüpfen wir dabei an Satz 4.18 an. Gemäß dieses Satzes kann jede IIM o.B.d.A. als konsistent vorausgesetzt werden. Allerdings bedingt die dort vorgestellte Transformation einer beliebigen IIM in eine konsistente die Lösung des Wortproblems für alle im Text bereits gelesenen Wörter. Dadurch kann sich aber der Aufwand an auszuführenden Berechnungen deutlich erhöhen. Ein besonders illustratives Beispiel ist der inkonsistent und iterativ arbeitende Algorithmus von Lange und Wiehagen (1991) zum exakten Lernen der Familie aller Patternsprachen. Dieser inkonsistente Algorithmus inferiert die Familie aller Patternsprachen überdies schwach-monoton und arbeitet bei jeder Eingabe in einer Zeit, die polynomiell in Abhängigkeit von der Länge der zu verarbeitenen Eingabe ist. In Wiehagen und Zeugmann (1992) konnten wir zeigen, daß es keinen konsistenten, in Polynomialzeit arbeitenen Lernalgorithmus zum exakten Lernen der Patternsprachen gibt, vorausgesetzt $\mathcal{P} \neq \mathcal{NP}$. Weitere Untersuchungen sollten sich mit dem Problem beschäftigen, unter welchen Bedingungen der Verzicht auf Konsistenz zu deutlich effizienteren Lernalgorithmen führt. Analoge Fragestellungen bieten sich natürlich auch hinsichtlich der einfgeführten Monotonieforderungen an.

5. Literaturverzeichnis

[1] Adleman, L.M., und Blum,M. (1991), Inductive inference and unsolvability, *Journal of Symbolic Logic* **56**, 891 - 900.

[2] Angluin, D. (1980a), Inductive inference of formal languages from positive data, *Information and Control* **45**, 117 - 135.

[3] Angluin, D. (1980b), Finding patterns common to a set of strings, *Journal of Computer and System Sciences* **21**, 46 - 62.

[4] Angluin, D. (1992), Computational learning theory: Survey and selected bibliography, *in* "Proceedings 24th Annual ACM Workshop on Theory of Computing," pp. 351 - 369, ACM Press.

[5] Angluin, D., und Smith, C.H. (1983), Inductive inference: theory and methods, *Computing Surveys* **15**, 237 - 269.

[6] Angluin, D., und Smith, C.H. (1987), Formal inductive inference, *in* "Encyclopedia of Artificial Intelligence" (St.C. Shapiro, Ed.), Vol. 1, pp. 409 - 418, Wiley-Interscience Publication, New York.

[7] Apsitis, K., Freivalds, R., Krikis, M., Simanovskis, R., und Smotrovs, J. (1992), Unions of identifiable classes of total recursive functions, *in* "Proceedings 3rd International Workshop on Analogical and Inductive Inference, October 1992, Dagstuhl Castle," (K.P. Jantke, Ed.) Lecture Notes in Artificial Intelligence Vol. 642, pp. 99 - 107, Springer-Verlag, Berlin.

[8] Barzdin, Ya.M. (1974a), Inductive inference of automata, functions and programs, *in* "Proceedings International Congress of Math.," Vancouver, pp. 455 - 460.

[9] Barzdin, Ya.M. (1974b), Две теоремы о предельном синтезе функций, *in* "Теория Алгоритмов и Программ," Vol. 1, (Ya.M. Barzdin, Ed.), pp.82 - 88, Latvian State University.

[10] Barzdin, Ya.M. (Ed.) (1974, 1975, 1977), "Теория Алгоритмов и Программ," Vol.1, 2, 3, Latvian State University.

[11] Barzdin, Ya.M., und Freivalds, R.V. (1972), On the prediction of general recursive functions, *Sov. Math. Dokl.* **13**, 1224 - 1228.

[12] Barzdin, Ya.M., und Freivalds, R.V. (1974), Прогнозирование и предельный синтез еффективно перечислимых классов функций, *in* "Теория Алгоритмов и Программ," Vol. 1 (Ya. M. Barzdin, ed.) Latvian State University, Riga, pp. 101 - 111.

[13] Blum, M. (1967) A machine-independent theory of the complexity of recursive functions, *Journal Association for Computing Machinery* **14**, 322 - 336.

[14] Blum, L., und Blum, M. (1975), Toward a mathematical theory of inductive inference, *Information and Control* **28**, 122 - 155.

[15] Board, R., und Pitt, L. (1992), On the necessity of Occam algorithms, *Theoretical Computer Science* **100**, 157 - 184.

[16] Brewka, G. (1991), "Nonmonotonic Reasoning: Logical Foundations of Commonsense," Cambridge University Press, Cambridge.

[17] Bucher, W., und Maurer, H. (1984), "Theoretische Grundlagen der Programmiersprachen, Automaten und Sprachen," Bibliographisches Institut AG, Wissenschaftsverlag, Zürich.

[18] Case, J. (1988), The power of vacillation, *in* "Proceedings 1st Annual ACM Workshop on Computational Learning Theory," (D. Haussler and L. Pitt, Eds.), pp. 196 -205, Morgan Kaufmann Publishers Inc.

[19] Case, J., und Lynes, C. (1982), Machine inductive inference and language identification, *in* "Proceedings Automata, Languages and Programming, Ninth Colloquium, Aarhus, Denmark," (M. Nielsen and E.M. Schmidt, Eds.), Lecture Notes in Computer Science Vol. 140, pp. 107 - 115, Springer-Verlag, Berlin.

[20] Case, J., und Smith, C.H. (1983), Comparison of identification criteria for machine inductive inference, *Theoretical Computer Science* **25**, 193 - 220.

[21] Chen, K.J. (1981), Tradeoffs in machine inductive inference, Ph. D. thesis, SUNY, Buffalo, NY.

[22] Chen, K.J. (1982), Tradeoffs in the inductive inference of nearly minimal size programs, *Information and Control* **52**, 68 - 86.

[23] Daley, R. (1983), On the error correcting power of pluralism in BC–type inductive inference, *Theoretical Computer Science* **24**, 95 - 104.

[24] Daley, R. (1986), Towards the development of an analysis of learning algorithms, *in* "Proceedings of the International Workshop of Analogical and Inductive Inference, Wendisch-Rietz, October 1986," (K.P. Jantke, Ed.), pp. 1 - 18, Lecture Notes in Computer Science, Vol. 265, Springer-Verlag, Berlin.

[25] Freivalds, R. (1990), Inductive inference of minimal programs, *in* "Proceedings 3rd Annual ACM Workshop on Computational Learning Theory," (M.A. Fulk and J. Case, Eds.), pp. 3 - 20, Morgan Kaufmann Publishers Inc.

[26] Freivalds, R., Kinber, E.B., und Wiehagen, R. (1992), Convergently versus divergently incorrect hypotheses in inductive inference, GOSLER Report 02/92, January 1992, Fachbereich Mathematik und Informatik, TH Leipzig.

[27] Freivalds, R., Kinber, E.B., und Wiehagen, R. (1993), On the power of inductive inference from good examples, *Theoretical Computer Science* **110**, 131 - 144.

[28] Fulk, M. (1990), Prudence and other restrictions in formal language learning, *Information and Computation* **85**, 1 - 11.

[29] Gasarch, W.I., und Smith, C.H. (1986), On the inference of sequences of functions, *in* "Proceddings International Workshop on Analogical and Inductive Inference, October 1986, Wendisch-Rietz," (K.P. Jantke, Ed.), Lecture Notes in Computer Science 265, pp. 23 - 41, Springer-Verlag.

[30] Gold, M.E. (1965), Limiting recursion, *Journal of Symbolic Logic* **30**, 28 - 48.

[31] Gold, M.E. (1967), Language identification in the limit, *Information and Control* **10**, 447 - 474.

[32] Grabowski, J. (1986), Ein Fortsetzungsprinzip in der Erkennungstheorie und seine Anwendung, *in* Strukturerkennung diskreter kybernetischer Systeme, (R. Linder, H. Thiele, Eds.), Seminarberichte der Sektion Mathematik der Humboldt-Universität Berlin Vol. 82.

[33] Helm, J.P. (1971), On effectively computable operators, *Zeitschrift f. mathematische Logik und Grundlagen der Mathematik (ZML)* **17**, 231 - 244.

[34] Jain, S., und Sharma, A. (1989), Recursion theoretic characterizations of language learning, The University of Rochester, Dept. of Computer Science, TR 281.

[35] Jantke, K.P. (1991a), Monotonic and non-monotonic inductive inference, *New Generation Computing* **8**, 349 - 360.

[36] Jantke, K.P. (1991b), Monotonic and non-monotonic inductive inference of functions and patterns, *in* "Proceedings First International Workshop on Nonmonotonic and Inductive Logic, December 1990, Karlsruhe," (J. Dix , K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 543, pp. 161 - 177, Springer-Verlag, Berlin.

[37] Jantke, K.P., und Beick, H.R. (1981), Combining postulates of naturalness in inductive inference, *Journal of Information Processing and Cybernetics (EIK)* **17**, 465 - 484.

[38] Jiang, T., Salomaa, A., Salomaa, K. und Yu, S. (1993), Inclusion is undecidable for pattern languages, *in* "Proceedings 20th International Colloquium on Automata, Languages and Programming, July 1993, Lund, Sweden," (A. Lingas, R. Karlsson, S. Carlsson, Eds.), Lecture Notes in Computer Science 700, pp. 301 - 312, Springer-Verlag, Berlin.

[39] Kapur, S. (1992), Monotonic language learning, *in* "Proceedings 3rd Workshop on Algorithmic Learning Theory, Tokyo, Japan, October 1992," JSAI, pp. 147 - 158.

[40] Kapur, S., und Bilardi, G. (1992), Language learning without overgeneralzation, *in* "Proceedings 9th Annual Symposuim on Theoretical Aspects of Computer Science, Cachan, France, February 13 - 15," (A. Finkel and M. Jantzen, Eds.), Lecture Notes in Computer Science Vol. 577, pp. 245 - 256, Springer-Verlag, Berlin.

[41] Kinber, E.B. (1993), persönliche Mitteilung.

[42] Kinber, E.B., und Zeugmann, T. (1985), Inductive inference of almost everywhere correct programs by reliably working strategies, *Journal of Information Processing and Cybernetics* **21**, 91 - 100.

[43] Kinber, E.B., und Zeugmann, T. (1991), One-sided error probabilistic inductive inference and reliable frequency identification, *Information and Computation* **92**, 253 - 284.

[44] Klette, R., und Wiehagen, R. (1980), Research in the theory of inductive inference by GDR mathematicians–A survey, *Information Science* **22**, 149 - 169.

[45] Kodratoff, Y., und Michalski, R.S. (1990), "Machine Learning, An Artificial Intelligence Approach," Vol. 3, Morgan Kaufmann Publ.

[46] Lange, S., und Wiehagen, R. (1991), Polynomial-time inference of arbitrary pattern languages, *New Generation Computing* **8**, 361 - 370.

[47] Lange, S., und Zeugmann, T. (1992a), Monotonic versus non-monotonic language learning, *in* "Proceedings 2nd International Workshop on Nonmonotonic and Inductive Logic, December 1991, Reinhardsbrunn," (G. Brewka, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence 659, pp. 254 - 269, Springer-Verlag, Berlin.

[48] Lange, S., und Zeugmann, T. (1992b), On the power of monotonic language learning, GOSLER-Report 05/92, February 1992, Fachbereich Mathematik und Informatik, TH Leipzig.

[49] Lange, S., und Zeugmann, T. (1992c), Types of monotonic language learning and their characterization, *in* "Proceedings 5th Annual ACM Workshop on Computational Learning Theory," Pittsburgh, July 1992, pp. 377 - 390, ACM Press, New York.

[50] Lange, S., und Zeugmann, T. (1992d), Learning recursive languages with bounded mind changes, GOSLER-Report 16/92, FB Mathematik und Informatik, TH Leipzig.

[51] Lange, S.,Zeugmann, T., und Kapur, S (1992), Class preserving monotonic language learning, submitted to *Theoretical Computer Science*, und GOSLER-Report 14/92, FB Mathematik und Informatik, TH Leipzig.

[52] Lange, S., und Zeugmann, T. (1993a), Language learning with a bounded number of mind changes, *in* "Proceedings 10th Annual Symposium on Theoretical Aspects of Computer Science, Würzburg, Februar 1993," (P. Enjalbert, A. Finkel and K.W. Wagner, Eds.), Lecture Notes in Computer Science 665, pp. 682 - 691, Springer-Verlag, Berlin.

[53] Lange, S., und Zeugmann, T. (1993b), Characterization of language learning from informant under various monotonicity constraints, erscheint in *Journal of Experimental and Theoretical Artificial Intelligence.*

[54] Lange, S., und Zeugmann, T. (1993c), Language learning in dependence on the space of hypotheses, *in* "Proceedings 6th Annual ACM Conference on Computational Learning Theory," Santa Cruz, July 1993, pp. 127 - 136, ACM Press, New York.

[55] Laudan, L. (1977), "Progress and Its Problems," University of California Press, Berkeley, California.

[56] Lewis, F.D. (1971), The enumerabilty and invariance of complexity classes, *Journal of Computer and System Sciences* **5**, 286 - 303.

[57] Strukturerkennung diskreter kybernetischer Systeme I, II, (1986), (R. Lindner, H. Thiele, Eds.), Seminarberichte der Sektion Mathematik der Humboldt-Universität Berlin Vol. 82.

[58] Machtey, M., und Young, P. (1978) "An Introduction to the General Theory of Algorithms," North-Holland, New York.

[59] Meyer, A.R., und Fischer, P.C. (1968), On computational speed-up. *in* "Proceedings 9th Annual IEEE Symposium on Switching and Automata Theory," pp. 351 - 355.

[60] Michalski, R.S., Carbonell, J.G., und Mitchell, T.M. (1984), "Machine Learning, An Artificial Intelligence Approach," Vol. 1, Springer-Verlag, Berlin.

[61] Michalski, R.S., Carbonell, J.G., und Mitchell, T.M. (1986), "Machine Learning, An Artificial Intelligence Approach," Vol. 2, Morgan Kaufmann Publ.

[62] Minicozzi, E. (1976), Some natural properties of strong-identification in inductive inference, *Theoretical Computer Science* **2**, 345 - 360.

[63] Mukouchi, Y. (1992a), Characterization of finite identification, *in* "Proceedings 3rd International Workshop on Analogical and Inductive Inference, October 1992, Dagstuhl Castle," (K.P. Jantke, Ed.) Lecture Notes in Artificial Intelligence Vol. 642, pp. 260 - 267, Springer-Verlag, Berlin.

[64] Mukouchi, Y. (1992b), Inductive Inference with Bounded Mind Changes, *in* Proceedings "3rd Workshop on Algorithmic Learning Theory, Tokyo,Japan, October 1992," JSAI, pp. 125 - 134.

[65] Natarajan, B.K. (1991), "Machine Learning," Morgan Kaufmann Publishers Inc.

[66] Nix, R. (1983), Editing by example, Ph.D thesis, Computer Science Department, Yale University, New Haven, Conn.

[67] Nix, R. (1984), Editing by example, *in* "Proceedings 11th ACM Workshop on Principles of Programming Languages, Salt Lake City, January 1984, pp. 186 - 195, ACM Press New York.

[68] William of Occam (circa 1320), Quodlibeta Septem (in translation).

[69] Osherson, D., Stob, M., und Weinstein, S. (1986), "Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists," MIT-Press, Cambridge, Massachusetts.

[70] Pitt, L. (1989a), A characterization of probabilistic inference, *Journal Association for Computing Machinery* **36**, 383 - 433.

[71] Pitt, L. (1989b), Inductive inference, dfas, and computational complexity, *in* "Proceedings International Workshop on Analogical and Inductive Inference, October 1989, Reinhardsbrunn Castle," (K.P. Jantke, Ed.), Lecture Notes in Artificial Intelligence 397, pp. 18 - 44, Springer-Verlag.

[72] Pitt, L., und Smith, C.H. (1988), Probabilty and plurality for aggregations of learning machines, *Information and Computation* **77**, 77 - 92.

[73] Podnieks, K.M. (1974), Сравнение различных типов синтеза и прогнозирования функций I, *in* "Теория Алгоритмов и Программ," Vol. 1 (Ya.M. Barzdin, Ed.), pp. 68 - 81, Latvian State University.

[74] Podnieks, K.M. (1975), Сравнение различных типов синтеза и прогнозирования функций II, *in* "Теория Алгоритмов и Программ," Vol. 2 (Ya.M. Barzdin, Ed.), pp. 35 - 44, Latvian State University.

[75] Popper, K. (1968), "The Logic of Scientific Discovery," Harper Torch Books.

[76] Porat, S., und Feldman, J.A. (1988), Learning automata from ordered examples, *in* "Proceedings First Annual ACM Workshop on Computational Learning Theory," (D. Haussler and L. Pitt, Eds.), pp. 386 - 396, Morgan Kaufmann Publishers Inc.

[77] Putnam, H. (1975), "Probability and Confirmation," Cambridge University Press.

[78] Rogers, H.Jr. (1967), "Theory of Recursive Functions and Effective Computability," McGraw-Hill, New York.

[79] Royer, J.S. (1986), Inductive inference of approximations, *Information and Control* **70**, 156 - 178.

[80] Shapiro, E. (1982), Algorithmic program debugging, Ph.D thesis, Computer Science Department, Yale University, New Haven, Conn.

[81] Shinohara, T. (1982), Polynomial time inference of pattern languages and its applications, *in* "Proceedings 7th ACM Symposium on Mathematical Foundations of Computer Science," pp. 193 - 208.

[82] Smith, C.H. (1982), The power of pluralism for automatic program synthesis, *Journal Association for Computing Machinery* **29**, 1144 - 1165.

[83] Solomonoff, R. (1964), A formal theory of inductive inference, *Information and Control* **7**, 1 - 22, 234 - 254.

[84] Trakhtenbrot, B.A., und Barzdin, Ya.M. (1970) "Конечные Автоматы (Поведение и Синтез)," Наука, Москва,
English translation: "Finite Automata-Behavior and Synthesis, Fundamental Studies in Computer Science 1," North-Holland, Amsterdam, 1973.

[85] Valiant, L.G. (1984), A theory of the learnable, *Communications of the ACM* **27**, 1134 - 1142.

[86] Wiehagen, R. (1976), Limes-Erkennung rekursiver Funktionen durch spezielle Strategien, *Journal Information Processing and Cybernetics (EIK)* **12**, 93 - 99.

[87] Wiehagen, R. (1977), Identification of Formal Languages, *in* "Proceedings Mathematical Foundations of Computer Science, Tatranska Lomnica," (J. Gruska, Ed.), Lecture Notes in Computer Science Vol. 53, pp. 571 - 579, Springer-Verlag, Berlin.

[88] Wiehagen, R. (1978), Zur Theorie der algorithmischen Erkennung, Dissertation B, Berlin.

[89] Wiehagen, R. (1991), A thesis in inductive inference, *in* "Proceedings First International Workshop on Nonmonotonic and Inductive Logic, December 1990, Karlsruhe," (J. Dix, K.P. Jantke and P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 543, pp. 184 - 207, Springer-Verlag, Berlin.

[90] Wiehagen, R. (1992), persönliche Mitteilung.

[91] Wiehagen, R., und Liepe, W. (1976), Charakteristische Eigenschaften von erkennbaren Klassen rekursiver Funktionen, *Journal of Information Processing and Cybernetics (EIK)* **12**, 421 - 438.

[92] Wiehagen, R., Freivalds, R., und Kinber, E.B. (1984), On the power of probabilistic strategies in inductive inference, *Theoretical Computer Science* **28**, 111 - 133.

[93] Wiehagen, R., and Zeugmann, T. (1992), Too much information can be too much for learning efficiently, *in* "Proceedings 3rd International Workshop on Analogical and Inductive Inference, October 1992, Dagstuhl Castle," (K.P. Jantke, Ed.) Lecture Notes in Artificial Intelligence Vol. 642, pp. 72 - 86, Springer-Verlag, Berlin.

[94] Zeugmann, T. (1983), A-posteriori characterizations in inductive inference of recursive functions, *Journal of Information Processing and Cybernetics (EIK)* **19**, 559 - 594.

[95] Zeugmann, T. (1984), On the nonboundability of total effective operators, *Zeitschrift f. mathematische Logik und Grundlagen der Mathematik (ZML)* **30**, 169 - 172.

[96] Zeugmann, T., Lange, S., und Kapur, S. (1992), Characterizations of class preserving monotonic and dual monotonic language learning, Technical Report IRCS 92 - 24, Institute for Research in Cognitive Science, University of Pennsylvania, Philadelphia.