Rekursiv aufzählbare Sprachen mit guten Beispielen lernen

...

26. Juni 1996

Zusammenfassung

Es wird das Lernen uniform rekursiv aufzählbarer Sprachfamilien anhand guter Beispiele untersucht und Unterschiede und Gemeinsamkeiten zum Lernen von rekursiven Sprachfamilien und rekursiven Funktionen aufgezeigt.

Dem verwendeten Modell liegt das Lernen von Schülern mit einem Lehrer zugrunde. Es werden verschiedene Varianten vorgestellt, verglichen und teilweise auch charakterisiert, und versucht, mit Beispielen und anderen typischen Eigenschaften ein Gefühl für die Leistungsfähigkeit zu vermitteln. Unter anderem wird gezeigt, daß es nicht immer „universelle“ gute Beispiele gibt, mit denen eine Sprachklasse in allen Situationen erklärt werden kann.

Inhaltsverzeichnis

Kapitel 1

Einleitung und Definitionen

Oft beginnen einer Erklärung dienende Sätze mit „Wenn zum Beispiel...“. In der Regel wird dieses Beispiel dann so gewählt, daß es das Problem möglichst gut darstellt. Wir verschwenden normalerweise keinen Gedanken an diese Art der Erklärung, so häufig begegnet sie uns. Dabei ist es oft erstaunlich, wie komplizierte Sachverhalte mit nur wenigen *guten* Beispielen erklärt werden können.

Diesen Ansatz, etwas anhand sorgfältig ausgewählter Beispiele zu lernen, haben Freivalds, Kinber und Wiehagen in [FKW a] und [FKW b] formalisiert und auf das Problem angewendet, rekursive Funktionen zu erlernen. Dabei haben sie unter anderem herausgefunden, daß sich mit endlich vielen guten Beispielen mehr lernen läßt, als mit *allen* Beispielen. Dieses recht erstaunliche Phänomen hat dazu geführt, das Modell auch auf andere Bereiche anzuwenden und resultierte unter anderen in einer Arbeit von Lange, Nessel und Wiehagen [LNW] zum Lernen von indizierten Sprachfamilien anhand guter Beispiele. Es ergaben sich eine Reihe interessanter und manche unerwartete Resultate. Der folgende Text untersucht nun, inwieweit sich die Resultate und Phänomene beim Lernen rekursiver Funktionen und indizierter Sprachfamilien auf das Lernen von uniform rekursiv aufzählbaren Sprachklassen übertragen lassen.

Der Rest dieses Kapitels gibt die notwendigen Definitionen und versucht, die Idee und die formale Umsetzung des Modells näher zu erläutern. In Kapitel 2 wird das Lernen von Sprachen anhand guter Beispiele untersucht, die jedoch nur aus der jeweils zu erlernenden Sprache stammen dürfen. Die Fähigkeiten des Lernens anhand von Textbeispielen werden charakterisiert und mit Beispielen versucht, die Arbeitsweise der Lernalgorithmen zu verdeutlichen. In [LNW] wurde nachgewiesen, daß es in den dort verwendeten Hypothesenräumen stets möglich ist, zu entscheiden, ob zwei Sprachen gleich sind. Hier wird nun gezeigt, daß sich diese schöne Eigenschaft nicht auf Hypothesenräume für rekursiv aufzählbare Sprachen übertragen läßt. Schließlich wird in diesem Kapitel gezeigt, daß eine Repräsentation des Halteproblems für verschiedene Gödelnumerierungen verschieden schwer zu lernen ist. Weil Gödelnumerierungen normalerweise immer als „äquivalent“ behandelt werden, ist dieses Resultat etwas überraschend. Kapitel 3 untersucht den Einfluß rekursiver Strategien auf das Lernen mit Textbeispielen. Es zeigt sich, daß diese Forderung Einbußen in der Lernkraft mit sich bringt. In Kapitel 4 dürfen die Beispiele sowohl aus der Sprache als auch aus dem Komplement der Sprache stammen. Hier wird das Lernen aus positiven und negativen Beispielen im Limes vollständig und das finite Lernen aus diesen Beispielen in einigen Spezialfällen charakterisiert. Weiter wird gezeigt, daß es keine „universellen“ Beispiele gibt; d.h. ein Lehrer kann nicht einfach „die“ guten Beispiele für eine Sprache berechnen und sie in allen Hypothesenräumen verwenden. D.h. was gute Beispiele „gut macht“ hängt nicht nur von der zu lernenden Sprache, sondern auch stark von der Umgebung ab, in der gelernt werden soll. Kapitel 5 wertet das Modell noch einmal im Licht der erhaltenen Ergebnisse und nennt offene Fragen.

Nun zu den technischen Details und notwendigen Definitionen. Im folgenden sind i, j, k, m, n, x, y, z stets natürliche Zahlen, die Menge der natürlichen Zahlen wird auch $I\!N$ geschrieben. Für eine endliche Menge $M \subseteq I\!N$ ist $max\ M$ das maximale Element in M, und $|M|$ ist die Anzahl der Elemente in M. $\langle , \rangle$ ist eine bijektive, berechenbare Funktion von $I\!N^2$ nach $I\!N$. Sprachen sind beliebige Mengen von endlichen Worten über einem endlichen Alphabet und werden normalerweise mit L bezeichnet. $\overline{L}$ ist das Komplement von L. Mögen A, B immer *endliche* Mengen von Wörtern bezeichnen; w ist immer ein endliches Wort. Es ist möglich, die Menge der endlichen Wörter über einem endlichen Alphabet berechenbar und bijektiv auf $I\!N$ abzubilden. Deshalb werden manchmal Worte und Zahlen synonym verwendet, je nachdem, was gerade zweckmäßiger ist. Außerdem induziert diese Abbildung eine Aufzählung $w_0, w_1, w_2, \dots$ aller endlichen Wörter über einem gewählten Alphabet.

Die Gesamtheit der berechenbaren – oder partiell rekursiven – Funktionen von $I\!N$ nach $I\!N$ wird mit $\mathcal{P}$ bezeichnet; die Menge der überall definierten – oder rekursiven – Funktionen in $\mathcal{P}$ heißt $\mathcal{R}$. Weiter ist $\mathcal{P}^n$ die Menge der n-stelligen berechenbaren Funktionen von $I\!N^n \mapsto I\!N$. $\mathcal{R}^n$ ist die Erweiterung von $\mathcal{R}$ auf n Stellen. Eine berechenbare Folge $\varphi_0, \varphi_1, \dots$ partiell rekursiver Funktionen wird Numerierung genannt. Formal ist eine Numerierung eine zweistellige partiell rekursive Funktion $\varphi(i, x)$ und es wird $\lambda x.\varphi(i, x)$ mit φ_i identifiziert. φ, ψ und ζ werden stets Numerierungen sein. Weiter bedeutet $\varphi_i(x)\downarrow$, daß φ_i an der Stelle x definiert und $\varphi_i(x)\uparrow$ entsprechend, daß φ_i an der Stelle x undefiniert ist. Mit K_φ ist stets die Menge $\{i \mid i \geq 0 \text{ und } \varphi_i(i)\downarrow\}$ gemeint. Falls φ aus dem Kontext hervorgeht oder ganz $\mathcal{P}$ aufzählt, wird der Index φ manchmal weggelassen. Für $f \in \mathcal{P}$ ist *domain* f der Definitions- und *range* f der Bildbereich von f. Manchmal werden Funktionen mit der Folge ihrer Werte identifiziert. So beschreibt z.B. $2\,0^\infty$ die Funktion

$$f(x) = \begin{cases} 2 & x = 0 \\ 0 & x > 0. \end{cases}$$

Analog beschreibt 1^n die endliche Funktion, die für 0 bis $n-1$ definiert ist und den Wert 1 ausgibt, für alle anderen Argumente ist diese Funktion undefiniert. Schließlich ist 1^n0 die endliche Funktion, die entsteht, wenn 1^n an der Stelle n auf 0 gesetzt wird. Einzelheiten können z.B. in [Rogers] nachgelesen werden. Manchmal werden auch Gödelnumerierungen verwendet. Damit sind zulässige Numerierungen im Sinne von [Rogers] gemeint. Sei Φ ein Blumsches Komplexitätsmaß für φ; siehe [Blum]. Für den vorliegenden Text ist es völlig ausreichend, sich Φ als Schrittzahlfunktion vorzustellen.

1.1 Definition. Sei ein beliebiges endliches Alphabet gegeben und für alle i sei L_i eine Menge von Wörtern über diesem Alphabet.

1) $\mathcal{L} = (L_i)_{i\geq 0}$ heißt *uniform rekursiv aufzählbar*, wenn es eine Funktion $\Lambda \in \mathcal{P}^2$ gibt, so daß für alle $i \in I\!N$ und alle Wörter w gilt:

$$\Lambda(i,w) = \begin{cases} 1 & w \in L_i, \\ 0 \text{ oder } \uparrow & w \notin L_i. \end{cases}$$

$\mathcal{L}$ heißt dann auch *Aufzählung* oder *Numerierung (der Sprachen L_i)*.

2) $\mathcal{RA} = \{\mathcal{L} \mid \mathcal{L} = (L_i)_{i\geq 0}$ ist uniform rekursiv aufzählbar$\}$

3) Für zwei Numerierungen $\mathcal{L} = (L_i)_{i\geq 0}$ und $\mathcal{L}' = (L'_i)_{i\geq 0}$ gilt genau dann $\mathcal{L} = \mathcal{L}'$, wenn $\{L_i \mid i \in I\!N\} = \{L'_i \mid i \in I\!N\}$; d.h. wenn beide Numerierungen die gleichen Sprachen aufzählen.

4) Analog ist $\mathcal{L} \subseteq \mathcal{L}'$ definiert. $\mathcal{L}'$ wird dann auch *umfassende Aufzählung für $\mathcal{L}$* genannt. •

Im weiteren Verlauf des Textes werden umfassende Numerierungen meistens ohne ausdrücklichen Hinweis verwendet. Die jeweiligen Stellen sind aber immer aus dem Kontext ersichtlich.

Indizierte Sprachfamilien sind ein Sonderfall der r.a. Sprachfamilien; bei ihnen ist Λ überall definiert. Die Menge aller indizierten Sprachfamilien wird mit $\mathcal{IF}$ bezeichnet. $\mathcal{L}$ mit diversen Indizes steht stets für uniform rekursiv aufzählbare Sprachklassen.

Definition 1.1 ist offensichtlich äquivalent zu der üblichen Definition von uniform rekursiv aufzählbar, bei der Λ entweder 1 oder undefiniert ist.

Des weiteren ist es oft vorteilhaft, Sprachen mit ihren (partiell-)charakteristischen Funktionen zu identifizieren. So ist etwa 1^∞ die Allsprache; 1^n die Sprache, die genau die ersten n Wörter enthält; 010101... enthält w_1, w_3, w_5, u.s.w.

1.2 Definition. Eine Familie $(ex_i)_{i\geq 0}$ von endlichen Wortmengen heißt *rekursiv generierbar (r.g.)*, wenn es ein berechenbares Verfahren gibt, das bei Eingabe eines beliebigen i die Menge ex_i ausgibt und stoppt. •

D.h. für jedes i kann eine Menge ex_i so berechnet werden, daß erkennbar ist, wann alle Elemente aus ex_i vorliegen.

1.3 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i\geq 0}$ eine Aufzählung von $\mathcal{L}'$.

1) $\mathcal{L}'$ heißt *anhand guter Textbeispiele finit bzgl. $\mathcal{L}$ erlernbar*, wenn es $S \in \mathcal{P}$ und eine r.g. Familie $(ex_i)_{i\geq 0}$ gibt, so daß für alle i gilt:

1. $ex_i \subseteq L_i$,
2. für alle endlichen $A \subseteq L_i$ existiert ein j mit $S(ex_i \cup A) = j$ und $L_j = L_i$.

Dies wird kurz $\mathcal{L}' \in$ TG-FIN$_\mathcal{L}$ geschrieben.

2) TG-FIN $= \{\mathcal{L}' \mid$ Es existiert eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$, so daß $\mathcal{L}'$ anhand guter Textbeispiele finit bzgl. $\mathcal{L}$ erlernbar ist.$\}$ •

„T“ soll an „Text“ erinnern, „G“ an „gute“ Beispiele. Die guten Beispiele dürfen also nur Worte aus der jeweiligen Sprache enthalten. TG-FIN ist nur für den Fall definiert, in dem eine uniform rekursiv aufzählbare Sprachklasse gelernt werden soll, im Gegensatz zur üblichen Induktiven Inferenz. Dort wird normalerweise immer eine Menge von rekursiven Funktionen als Lernziel vorgegeben, unaghängig davon, ob diese Menge numerierbar ist. Mit anderen Worten, die verwendeten Hypothesenräume sind immer umfassend. Definition 1.3 läßt keine Aussagen über $\mathcal{M} = \{1^i 0^\infty \mid i \in \overline{M}\}$ zu, wenn $M \subseteq \mathbb{N}$ rekursiv aufzählbar, aber nicht rekursiv ist, weil $\mathcal{M}$ natürlich keine Numerierung besitzt, die *genau* die Sprachen dieser Menge aufzählt. Die Formulierung von 1.3 ist so gewählt, daß sie möglichst der Definition von TG-FIN für *indizierte* Sprachklassen aus [LNW] ähnelt und die Resultate vergleichbar bleiben.

Als Hypothesenräume sind bei TG-FIN nur Umnumerierungen der gegebenen Klasse $\mathcal{L}'$ erlaubt. Später werden die Inferenztypen CTG-FIN und CTG-LIM eingeführt, die auch umfassende Numerierungen von $\mathcal{L}'$ als Hypothesenräume zulassen.

Kurz zum Modell: Ein Lehrer berechnet effektiv die endlich vielen guten Beispiele und gibt diese an die Strategie S weiter, die einen Schüler modellieren soll. Die Strategie bekommt außerdem noch eine endliche Menge A zu den berechneten guten Beispielen. Dies geschieht in erster Linie, um Codierungstricks vorzubeugen, die zu einer trivialen Lernbarkeit aller rekursiv aufzählbaren Klassen führen würden. Es wäre ggf. möglich, genau i Worte als gute Beispiele zu wählen, wenn die Sprache L_i gelernt werden soll.

In der Realität ist es zudem so, daß nur selten wirklich genau die notwendigen und hinreichenden Beispiele gefunden werden. Es gilt hier oft die Regel „lieber eins zuviel als eins zuwenig“. Nun sollten aber ein paar Beispiele mehr die Strategie nicht am Lernen hindern, wenn auch die „richtigen“ Beispiele präsentiert werden.

1.4 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i \geq 0}$ eine Aufzählung von $\mathcal{L}'$.

1) $\mathcal{L}'$ heißt *anhand guter Textbeispiele im Limes bzgl. $\mathcal{L}$ erlernbar*, wenn es $S \in \mathcal{P}^2$ und eine r.g. Familie $(ex_i)_{i \geq 0}$ gibt, so daß für alle i gilt:

1. $ex_i \subseteq L_i$,
2. für alle endlichen $A \subseteq L_i$ existiert ein j mit $\lim_{n \to \infty} S(ex_i \cup A, n) = j$ und $L_j = L_i$.

Dies wird kurz $\mathcal{L}' \in$ TG-LIM$_\mathcal{L}$ geschrieben.

2) TG-LIM $= \{\mathcal{L}' \mid$ Es existiert eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$, so daß $\mathcal{L}'$ anhand guter Textbeispiele im Limes bzgl. $\mathcal{L}$ erlernbar ist.$\}$ •

Das Informationsangebot ist das gleiche wie bei TG-FIN, nur darf der Schüler hier seine Hypothese endlich oft wechseln. Auf den ersten Blick erscheint es unnötig, der Strategie Hypothesenwechsel zu erlauben, wenn sie doch nur einmal Informationen bekommt. Wie sich beim Lernen rekursiver Funktionen anhand guter Beispiele gezeigt hat, ist es aber manchmal unumgänglich, sich zeitweise auf falsche Vermutungen einzulassen.

Beim Lernen aus Informantbeispielen darf der Lehrer sowohl Beispiele aus der Sprache als auch aus dem Komplement der Sprache präsentieren. Das ist bei rekursiv aufzählbaren Sprachen ein bißchen problematisch, denn die Komplemente der Sprachen sind im allgemeinen nicht rekursiv aufzählbar.

1.5 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i \geq 0}$ eine Aufzählung von $\mathcal{L}'$.

1) $\mathcal{L}'$ heißt *anhand guter Informantbeispiele finit bzgl. $\mathcal{L}$ erlernbar*, wenn es $S \in \mathcal{P}^2$ und r.g. Familien $(ex_i^+)_{i \geq 0}$ und $(ex_i^-)_{i \geq 0}$ gibt, so daß für alle i gilt:

1. $ex_i^+ \subseteq L_i$ und $ex_i^- \subseteq \overline{L_i}$,
2. für alle endlichen $A \subseteq L_i$ und $B \subseteq \overline{L_i}$ existiert ein j mit $S(ex_i^+ \cup A, ex_i^- \cup B) = j$ und $L_j = L_i$.

Kurzschreibweise: $\mathcal{L}' \in$ IG-FIN$_{\mathcal{L}}$.

2) IG-FIN $= \{\mathcal{L}' \mid$ Es existiert eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$, so daß $\mathcal{L}'$ anhand guter Informantbeispiele finit bzgl. $\mathcal{L}$ erlernbar ist.$\}$ •

1.6 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i \geq 0}$ eine Aufzählung von $\mathcal{L}'$.

1) $\mathcal{L}'$ heißt *anhand guter Informantbeispiele im Limes bzgl. $\mathcal{L}$ erlernbar*, wenn es $S \in \mathcal{P}^3$ und r.g. Familien $(ex_i^+)_{i \geq 0}$ und $(ex_i^-)_{i \geq 0}$ gibt, so daß für alle i gilt:

1. $ex_i^+ \subseteq L_i$ und $ex_i^- \subseteq \overline{L_i}$,
2. für alle endlichen $A \subseteq L_i$ und $B \subseteq \overline{L_i}$ existiert ein j mit $\lim_{n \to \infty} S(ex_i^+ \cup A, ex_i^- \cup B, n) = j$ und $L_j = L_i$.

2) IG-LIM $= \{\mathcal{L}' \mid$ Es existiert eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$, so daß $\mathcal{L}'$ anhand guter Informantbeispiele im Limes bzgl. $\mathcal{L}$ erlernbar ist.$\}$ •

1.7 Bezeichnung. Mit FINITE $= (E_i)_{i \geq 0}$ wird eine wiederholungsfreie, rekursiv generierbare Aufzählung aller endlichen Wortmengen über einem – jeweils aus dem Kontext hervorgehenden – endlichen Alphabet bezeichnet.

Nun einige Standarddefinitionen zum Lernen anhand von Text.

1.8 Definition. Sei L eine beliebige Sprache.

- Eine Folge $(t_i)_{i \geq 0}$ von Worten und „$*$“ heißt *Text für L*, falls $L = \{t_i \mid i \in \mathbb{N} \wedge t_i \neq *\}$ gilt. Die Menge aller Texte von L wird mit $Text(L)$ bezeichnet.
- t^n steht für die ersten n Elemente eines Textes t. •

1.9 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i \geq 0}$ eine Aufzählung von $\mathcal{L}'$. Dann erlernt eine Strategie $S \in \mathcal{P}$ die Sprachklase $\mathcal{L}'$ *im Limes bezüglich $\mathcal{L}$ anhand Text*, falls S für alle $L \in \mathcal{L}'$ und alle Texte $t \in Text(L)$ folgende Bedingungen erfüllt:

- $\forall n : S(t^n) \downarrow$.

- $m = \lim_{n \to \infty} S(t^n)$ existiert und $L_m = L$.

T-LIM $= \{\mathcal{L}' \mid$ Es gibt eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$ und eine Strategie S, die $\mathcal{L}'$ bezüglich $\mathcal{L}$ im Limes anhand Text erlernt$\}$ •

Und zum Abschluß noch die Definition von finitem Lernen anhand von Text.

1.10 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i \geq 0}$ eine Aufzählung von $\mathcal{L}'$. Dann erlernt eine Strategie $S \in \mathcal{P}$ die Sprachklasse $\mathcal{L}'$ *finit bezüglich* $\mathcal{L}$ *anhand Text*, falls S für alle $L \in \mathcal{L}'$ und alle Texte $t \in Text(L)$ folgende Bedingungen erfüllt:

- $\forall n : S(t^n) \downarrow$,
- $\exists n \forall m < n : S(t^m) =$ „?“ $\wedge\ S(t^n) = k\ \wedge\ L_k = L_i$.

T-FIN $= \{\mathcal{L}' \mid$ Es gibt eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$ und eine Strategie S, die $\mathcal{L}'$ bezüglich $\mathcal{L}$ finit anhand Text erlernt$\}$ •

Diese Inferenztypen sind gut bekannt, und sie wurden so oft begründet und erklärt, daß an dieser Stelle darauf verzichtet sei. Als Quelle für weitere Informationen seien [Gold] und [OSW] genannt.

Der Unterschied zwischen dem Lernen mit guten Beispielen und dem Lernen anhand von Text liegt vor allem im Informationsangebot. Bei den guten Beispielen ist diese Information immer endlich und wird auf *einmal* gegeben, während der Text für eine unendliche Sprache natürlich alle diese unendlich vielen Worte enthält und Stück für Stück präsentiert wird. D.h. eine Strategie, die mit guten Beispielen arbeitet, muß ihre Hypothesen bilden, obwohl sie „fast nichts“ von dieser unendlichen Sprache gesehen hat. Sie besitzt auch keine Möglichkeit, sich weitere Informationen zu beschaffen. „Textstrategien“ müssen zwar ebenfalls nach einem endlichen Textstück konvergieren, nur haben sie die Wahl, sich so lange den Text anzusehen, bis sie eine gewünschte Information gesehen haben. Ihr Informationsangebot ist also beliebig groß.

Kapitel 2

Lernen mit Textbeispielen

Als erstes werden die Inferenztypen TG-FIN und TG-LIM charakterisiert und anschließend ihre Möglichkeiten und Grenzen durch Beispiele und andere Eigenschaften ausgeleuchtet.

2.1 Satz. Sei $\mathcal{L}' \in \mathcal{RA}$. Es gilt $\mathcal{L}' \in$ TG-FIN(LIM) $\iff$ Es gibt eine Aufzählung $\mathcal{L}$ von $\mathcal{L}'$ und eine r.g. Familie $(ex_i)_{i\geq 0}$, so daß für alle i gilt:

1) $ex_i \subseteq L_i$,

2) $\forall j : ex_i \subseteq L_j \ \wedge \ ex_j \subseteq L_i \Rightarrow L_j = L_i$.

Im allgemeinen ist $\{(i,j) \mid ex_i \subseteq L_j\}$ rekursiv aufzählbar, aber nicht entscheidbar.

Beweis: „$\Rightarrow$" Sei also $\mathcal{L}' \in$ TG-FIN(LIM) bzgl. $\mathcal{L}$. Dann gibt es S_f und S_l (für FIN bzw. LIM) und r.g. Familien (ex_i^f) und (ex_i^l) mit den entsprechenden Eigenschaften. Dann gilt nach Definition von TG-FIN(LIM) sofort 1).

Seien nun i,j beliebig mit $ex_i^{f/l} \subseteq L_j$ und $ex_j^{f/l} \subseteq L_i$. Dann gilt für S_f nach Voraussetzung

$$L_i = L_{S_f(ex_i^f)} = L_{S_f(ex_i^f \cup ex_j^f)} = L_{S_f(ex_j^f)} = L_j.$$

Also $L_i = L_j$ und somit gilt 2), da i,j beliebig waren. Analog gilt für S_l

$$L_i = L_{\lim\limits_{n\to\infty} S_l(ex_i^l,n)} = L_{\lim\limits_{n\to\infty} S_l(ex_i^l \cup ex_j^l,n)} = L_{\lim\limits_{n\to\infty} S_l(ex_j^l,n)} = L_j,$$

und 2) folgt auch hier.

„$\Leftarrow$" Zuerst der Beweis für TG-LIM. Dieser Teil des Beweises ist formal nicht notwedig, erlaubt aber eine interessante Beobachtung. Definiere $S(A,n)$ wie folgt.

$$S(A,n) = \mu(z \leq n)[ex_z \subseteq A \subseteq L_z \text{ ist in } n \text{ Schritten beweisbar}, \uparrow \text{ sonst.}]$$

Offensichtlich gilt $S \in \mathcal{P}$. Sei i beliebig und gelte $ex_i \subseteq A \subseteq L_i$. Es existiert damit ein k, so daß $ex_i \subseteq A \subseteq L_i$ in k Schritten beweisbar ist. D.h. für alle $k' > k$ gilt

$S(A, k') \leq i$. Weiter ist $(S(A,n))_{n \geq k'}$ eine monoton fallende Folge, die nach unten durch 0 beschränkt ist: Wenn $ex_j \subseteq A \subseteq L_j$ in n Schritten beweisbar ist, kann S für alle $n' > n$ offensichtlich nichts Größeres als j ausgeben.

Also konvergiert $(S(A,n))_{n \geq k}$, z.B. gegen m. D.h. es gilt $ex_m \subseteq A \subseteq L_m$ und $ex_i \subseteq A \subseteq L_i$. Mit 2) folgt $L_m = L_i$. Weil i beliebig war, folgt die Behauptung.

Nun der Beweis für TG-FIN. Definiere $S(A)$ wie folgt.

$S(A)$ = „Suche ein z mit $ex_z \subseteq A \subseteq L_z$. Wenn eines gefunden wurde, gib es aus. Stop.“

Offensichtlich ist $S \in \mathcal{P}$. Sei wieder i beliebig und gelte $ex_i \subseteq A \subseteq L_i$. Also terminiert S, z.B. $S(A) = n$. Dann gilt $ex_i \subseteq A \subseteq L_i$ und $ex_n \subseteq A \subseteq L_n$. Also $ex_i \subseteq L_n$ und $ex_n \subseteq L_i$ und es folgt wegen 2) $L_i = L_n$. S lernt damit L_i. Weil i beliebig war, folgt die Behauptung. □

2.2 Korollar. TG-FIN = TG-LIM

Dazu einige Bemerkungen.

Das notwendige und hinreichende Kriterium aus Satz 2.1 entspricht *genau* dem für indizierte Sprachfamilien aus [LNW]. D.h. die Voraussetzung „$\mathcal{L}$ indizierte Sprachfamilie“ ist nicht notwendig.

Dieser Satz widerlegt die Vermutung „TG-FIN $\neq$ TG-LIM für nicht indizierte Sprachfamilien“, die ebenfalls in [LNW] geäußert wurde.

Allerdings gibt es zwischen TG-FIN und TG-LIM einen Unterschied, wie aus dem Beweis des Satzes deutlich wird: Alle von der Limesstrategie S ausgegebenen Hypothesen sind richtig, sie könnte also auch – ohne Einbuße der Lernkraft – nach der ersten Hypothese stoppen. Allerdings *optimiert* S diese Hypothese nun soweit es geht. Böse Zungen könnten nun behaupten, daß TG-FIN den „theoretischen“ Standpunkt beschreibt: „Wir haben eine richtige Erklärung, warum sich noch damit beschäftigen?“, während TG-LIM den „praktischen“ Aspekt beschreibt: „So, wir wissen, es ist möglich. Nun wollen wir doch 'mal sehen, wie gut es geht.“

Korollar 2.2 ist recht erstaunlich, da wieder alle „wesentlichen“ Informationen über r.a. Sprachfamilien aus TG-LIM schon finit gewonnen werden können, und das, obwohl nicht einmal „$w \in L_i$?“ uniform entscheidbar ist.

Es scheint dafür zwei Gründe zu geben.

1. Mittels der guten Beispiele bekommt S „finite“ Information über die L_i, die nur noch verifiziert werden muß.

2. S muß *alle* Sprachen aus $\mathcal{L}'$ lernen und für das Lernen von rekursiven Funktionen gilt ja bereits, daß jede numerierbare Menge rekursiver Funktionen in Gex-Fin liegt, siehe [FKW b], also insbesondere auch in Gex-Lim. (Siehe Definitionen 2.3 und 2.4.) Erst wenn die Strategie nicht mehr jede Funktion der Numerierung lernen muß, werden die Limesstrategien stärker.

Der folgende Satz besagt, daß auch für uniform r.a. Sprachklassen die Limesstrategien stärker werden, wenn in *umfassenden* Hypothesenräumen gelernt wird. Um das zu zeigen, sind allerdings weitere Definitionen nötig.

2.3 Definition. [FKW b]
1) Sei $U \subseteq \mathcal{R}$ und ψ eine beliebige Numerierung. U heißt *finit mit guten Beispielen bzgl.* ψ *erlernbar* $\iff$ Es gibt eine Numerierung ex, eine Strategie $S \in \mathcal{P}$ und eine Funktion $z \in \mathcal{P}$, so daß $U \subseteq \mathcal{P}_\psi$ und für alle i mit $\psi_i \in U$ gilt

(1) ex_i ist eine endliche Teilfunktion von ψ_i, $z(i)\downarrow$ und $z(i) = |domain\; ex_i|$,

(2) für jede endliche Teilfunktion A von ψ_i gilt $\psi_{S(ex_i \cup A)} = \psi_i$.

2) Gex-Fin$_\psi$ = $\{U \mid U$ ist im Limes mit guten Beispielen bzgl. ψ erlernbar $\}$

3) Gex-Fin = $\bigcup\limits_{\psi \in \mathcal{P}}$ Gex-Fin$_\psi$ •

2.4 Definition. [FKW b]
1) Sei $U \subseteq \mathcal{R}$ und ψ eine beliebige Numerierung. U heißt *im Limes mit guten Beispielen bzgl.* ψ *erlernbar* $\iff$ Es gibt eine Numerierung ex, eine Strategie $S \in \mathcal{P}^2$ und eine Funktion $z \in \mathcal{P}$, so daß $U \subseteq \mathcal{P}_\psi$ und für alle i mit $\psi_i \in U$ gilt

(1) ex_i ist eine endliche Teilfunktion von ψ_i, $z(i)\downarrow$ und $z(i) = |domain\; ex_i|$,

(2) für jede endliche Teilfunktion A von ψ_i gibt es ein j mit $\psi_i = \psi_j$ und $\lim\limits_{n\to\infty} S(ex_i \cup A, n) = j$.

2) Gex-Lim$_\psi$ = $\{U \mid U$ ist im Limes mit guten Beispielen bzgl. ψ erlernbar $\}$

3) Gex-Lim = $\bigcup\limits_{\psi \in \mathcal{P}}$ Gex-Lim$_\psi$ •

2.5 Satz. [FKW b] $\mathcal{R} \in$ Gex-Lim

Der nächste Satz steht nicht ausdrücklich in [FKW b], ist aber eine offensichtliche Folgerung der dort angegebenen Theoreme.

2.6 Satz. [FKW b] $\mathcal{R} \notin$ Gex-Fin

2.7 Definition. Sei $f \in \mathcal{P}$, $U \subseteq \mathcal{P}$, $\varphi \in \mathcal{P}^2$.
1) $L_f = \{\langle i, f(i)\rangle \mid i \in domain\; f\}$

2) $\mathcal{L}_U = \{L_f \mid f \in U\}$

3) $\mathcal{L}_\mathcal{R} = \{L_f \mid f \in \mathcal{R}\}$

4) $\mathcal{L}_\varphi = \{L_{\varphi_i} \mid i \in I\!N\}$

5) $\mathcal{L}_{\mathcal{R}\varphi} = \mathcal{L}_\varphi \cap \mathcal{L}_\mathcal{R}$ •

Offensichtlich gilt dann für ein beliebiges $\varphi \in \mathcal{P}^2$: $L_{\varphi_i} = L_{\varphi_j} \iff \varphi_i = \varphi_j$.

Definition. Sei $\mathcal{L}' \in \mathcal{RA}$, $\mathcal{L} = (L_i)_{i\geq 0}$ umfassende Aufzählung von $\mathcal{L}'$. $(ex_i)_{i\geq 0}$ heißt *rekursiv generierbar (r.g.) bzgl.* $(\mathcal{L}, \mathcal{L}')$, wenn ein Algorithmus existiert, der zu jedem i mit $L_i \in \mathcal{L}'$ eine endliche Menge $ex_i \subseteq L_i$ ausgibt und danach stoppt. •

Für i mit $L_i \notin \mathcal{L}'$ ist das Verhalten des Algorithmus nicht näher spezifiziert. Der Grund für diese Definition ist folgender. Der Lehrer muß nur für solche Sprachen effektiv gute Beispiele berechnen können, die auch wirklich in der zu erlernden Klasse $\mathcal{L}'$ liegen. Auf anderen Sprachen wird das Verhalten des Algorithmus nicht näher definiert, um die Wahl des Lehrers nicht einzuschränken.

Wenn $\mathcal{L} = \mathcal{L}'$ ist, dann entspricht die rekursive Generierbarkeit bzgl. $(\mathcal{L}, \mathcal{L}')$ der rekursiven Generierbarkeit aus Definition 1.2.

2.8 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i\geq 0}$ eine *umfassende* Aufzählung von $\mathcal{L}'$.

1) $\mathcal{L}'$ heißt *anhand guter Textbeispiele umfassend finit bzgl. $\mathcal{L}$ erlernbar*, wenn es $S \in \mathcal{P}$ und eine Familie $(ex_i)_{i\geq 0}$ gibt, so daß gilt:

1. $(ex_i)_{i\geq 0}$ ist r.g. bzgl. $(\mathcal{L}, \mathcal{L}')$,
2. für alle i mit $L_i \in \mathcal{L}'$ und alle endlichen $A \subseteq L_i$ existiert ein j mit $S(ex_i \cup A) = j$ und $L_j = L_i$.

2) CTG-FIN $= \{\mathcal{L}' \mid$ Es existiert eine *umfassende* Aufzählung $\mathcal{L}$ von $\mathcal{L}'$, so daß $\mathcal{L}'$ anhand guter Textbeispiele finit bzgl. $\mathcal{L}$ erlernbar ist.$\}$

3) $\mathcal{L}'$ heißt *anhand guter Textbeispiele umfassend im Limes bzgl. $\mathcal{L}$ erlernbar*, wenn es $S \in \mathcal{P}^2$ und eine Familie $(ex_i)_{i\geq 0}$ gibt, so daß gilt:

1. $(ex_i)_{i\geq 0}$ ist r.g. bzgl. $(\mathcal{L}, \mathcal{L}')$,
2. für alle i mit $L_i \in \mathcal{L}'$ und alle endlichen $A \subseteq L_i$ existiert ein j mit $\lim_{n\to\infty} S(ex_i \cup A, n) = j$ und $L_j = L_i$.

4) CTG-LIM $= \{\mathcal{L}' \mid$ Es existiert eine *umfassende* Aufzählung $\mathcal{L}$ von $\mathcal{L}'$, so daß $\mathcal{L}'$ anhand guter Textbeispiele im Limes bzgl. $\mathcal{L}$ erlernbar ist.$\}$ •

Das „C“ soll an „umfassend“ (engl. „comprising“) erinnern. Diesen Begriff haben Lange und Zeugmann eingeführt, siehe zum Beispiel [LZ a], [LZ b] und die darin enthaltenen Verweise. Die Analogie zu Definition 2.3 ist unverkennbar; vor allem „für alle i mit $L_i \in \mathcal{L}'$“ ist in dieser Definition der Definition 2.3 viel ähnlicher als „für alle i“ in Definition 1.3 und 1.4.

2.9 Lemma. $\mathcal{L}_\mathcal{R} \in$ CTG-LIM $\setminus$ CTG-FIN.

Beweis: Für $\varphi \in \mathcal{P}^2$ mit $\mathcal{R} \subseteq \mathcal{P}_\varphi$ gilt offensichtlich $\mathcal{L}_\mathcal{R} = \mathcal{L}_{\mathcal{R}\varphi}$. Also genügt es, die beiden folgenden Behauptungen zu zeigen:

1) $\exists \varphi \in \mathcal{P}^2 : \mathcal{L}_\mathcal{R} = \mathcal{L}_{\mathcal{R}\varphi} \wedge \mathcal{L}_{\mathcal{R}\varphi} \in$ CTG-LIM,

2) $\forall \varphi \in \mathcal{P}^2 : \mathcal{L}_\mathcal{R} = \mathcal{L}_{\mathcal{R}\varphi} \Rightarrow \mathcal{L}_{\mathcal{R}\varphi} \notin$ CTG-FIN.

Diese Behauptungen werden durch Reduktion auf Satz 2.5 bewiesen.

Zu 1): Nach Satz 2.5 gilt $\mathcal{R} \in$ Gex-Lim. Also existieren φ, S, ex und z wie in Definition 2.4 gefordert. Insbesondere gilt dann $\mathcal{R} \subseteq \mathcal{P}_\varphi$ und damit $\mathcal{L}_{\mathcal{R}\varphi} = \mathcal{L}_\mathcal{R}$. Definiere für alle i:

$$_{\mathcal{L}}ex_i = \{\langle x, \varphi_i(x)\rangle \mid (x, \varphi_i(x)) \in ex_i\}.$$

Für den Rest dieses Beweises wird statt L_{φ_i} kurz L_i geschrieben. Also ist $(_{\mathcal{L}}ex_i)$ r.g. und für alle i gilt $_{\mathcal{L}}ex_i \subseteq L_i$.
Sei $\pi_1(\langle x,y\rangle) = x$ und $\pi_2(\langle x,y\rangle) = y$ und weiter $A = \{w_1, \ldots, w_n\}$ eine Menge von Wörtern. Definiere eine Funktion *decod* wie folgt. (Dabei wird die Äquivalenz von Worten und $I\!N$ ausgenutzt.)

$$decod(A) = \{(\pi_1(w_i), \pi_2(w_i)) \mid w_i \in A\}$$

Definiere nun eine Strategie $\mathcal{L}S$ durch

$$\mathcal{L}S(A,n) = S(decod(A), n).$$

Damit ist $\mathcal{L}S \in \mathcal{P}^2$. Sei nun $L \in \mathcal{L}_{\mathcal{R}\varphi}$ beliebig, etwa $L = L_i$. Sei $f = \varphi_i$. Es gelten $decod(_{\mathcal{L}}ex_j) = ex_j$ und $A \subseteq L_j \Rightarrow decod(A) \subseteq \varphi_j$, für alle j, nach Definition von $_{\mathcal{L}}ex_i$ und *decod*. Aus $_{\mathcal{L}}ex_j \subseteq A \subseteq L_j$ folgt also $ex_j \subseteq decod(A) \subseteq \varphi_j$ für alle j, nach Definition von $\mathcal{L}_{\mathcal{R}\varphi}$ und $\mathcal{L}_\mathcal{R}$. Setze

$$m = \lim_{n\to\infty} \mathcal{L}S(A,n) = \lim_{n\to\infty} S(decod(A), n).$$

Da $f \in \mathcal{R}$, existiert m nach Voraussetzung. Aus der Definition von $\mathcal{L}_\varphi$ folgt $L_i = L_j \iff \varphi_i = \varphi_j$. Also $L_m = L_i$, da nach Voraussetzung $\varphi_m = \varphi_i$. $\mathcal{L}S$ erlernt also L. Da L beliebig war, folgt 1).

Zu 2): Angenommen, es gibt ein $\varphi \in \mathcal{P}^2$ mit $\mathcal{L}_\mathcal{R} = \mathcal{L}_{\mathcal{R}\varphi}$ und $\mathcal{L}_{\mathcal{R}\varphi} \in$ CTG-FIN. Dann gibt es eine Strategie $\mathcal{L}S$, gute Beispiele $(_{\mathcal{L}}ex_i)$ wie in der Definition von CTG-FIN gefordert. Definiere für alle i

$$\psi_i(x) = \begin{cases} \pi_2(w) & \exists w \in L_i : \pi_1(w) = x \\ \uparrow & \text{sonst.} \end{cases}$$

Dann ist $\psi \in \mathcal{P}^2$ und $\mathcal{R} \subseteq \mathcal{P}_\psi$ nach Definition von $\mathcal{L}_\mathcal{R}$ und $\mathcal{L}_{\mathcal{R}\psi}$. Aus der Definition folgt weiter $(*)$ $\psi_i = \psi_j \iff L_i = L_j$ und $\psi_i \in \mathcal{R} \iff L_i \in \mathcal{L}_\mathcal{R}$. Definiere nun für alle i

$$ex_i = \{(\pi_1(w), \pi_2(w)) \mid w \in {_{\mathcal{L}}ex_i}\}.$$

Damit ist (ex_i) r.g. (bzw. erfüllt die Bedingung (1) aus Definition 2.3). Sei $A \subseteq f \in \mathcal{P}$, $A = \{(n_1, m_1), \ldots, (n_k, m_k)\}$. Setze

$$cod(A) = \{\langle n_i, m_i\rangle \mid i = 1, \ldots, k\}.$$

Nun wird eine Strategie S wie folgt definiert:

$$S(A) = \mathcal{L}S(cod(A)).$$

Also ist $S \in \mathcal{P}$.
Sei nun ein $\varphi_i \in \mathcal{R}$ – und damit $L_i \in \mathcal{L}_{\mathcal{R}\varphi}$ – beliebig gewählt. Es gilt $cod(ex_i) = {}_{\mathcal{L}}ex_i$ für alle i nach Definition von cod und damit folgt aus ${}_{\mathcal{L}}ex_i \subseteq A \subseteq \psi_i$ unmittelbar ${}_{\mathcal{L}}ex_i \subseteq cod(A) \subseteq L_i$. Weil $L_i \in \mathcal{L}_{\mathcal{R}\varphi}$ ist, existiert

$$m = \mathcal{L}S(cod(A)) = S(A).$$

Nach Voraussetzung gilt $L_m = L_i$ und es folgt wegen $(*)$ sofort $\psi_m = \psi_i (= \varphi_i)$. Also erlernt S jede rekursive Funktion φ_i. D.h. $\mathcal{R} \in$ Gex-Fin$_\psi$ mit Strategie S. $\lightning$ zu Satz 2.6. $\square$

Es gibt eine enge Beziehung zwischen CTG-FIN und Gex-Fin, wie das folgende Lemma belegt. Für ein $U \subseteq \mathcal{P}$ hat die Menge $\mathcal{L}_U$ eine ganz spezielle Gestalt; siehe dazu Definition 2.7. In [Rogers] werden Mengen der Form L_f als „single-valued" bezeichnet. Es ist klar, daß nicht alle $\mathcal{L} \in$ TG-FIN Familien „eindeutig-wertiger" Mengen sind. In gewisser Weise ist also das Lernen von uniform rekursiv aufzählbaren Sprachfamilien eine Verallgemeinerung des Lernens rekursiver Funktionen.

2.10 Lemma. Sei $U \subseteq \mathcal{R}, \psi \in \mathcal{P}^2$. Dann gilt

$$U \in \text{ Gex-Fin}_\psi \iff \mathcal{L}_U \in \text{ CTG-FIN}_{\mathcal{L}_\psi}.$$

Beweis: In diesem Beweis findet die Notation aus dem Beweis von Lemma 2.9 Verwendung. Beachte, daß hier umfassende Aufzählungen verwendet werden.

„$\Rightarrow$": Sei also $U \in$ Gex-Fin$_\psi$ mit entsprechenden guten Beispielen $(ex_i)_{i\geq 0}$ und einer Strategie S. Wie in 2.9 definiere ${}_{\mathcal{L}}ex_i = cod(ex_i)$ für alle i und setze $\mathcal{L}S(A) = S(decod(A))$. Dann ist $({}_{\mathcal{L}}ex_i)_{i\geq 0}$ r.g. und $\mathcal{L}S$ berechenbar. Sei nun $L_i \in \mathcal{L}_U$ beliebig und gelte ${}_{\mathcal{L}}ex_i \subseteq A \subseteq L_i$. Es folgt $decod({}_{\mathcal{L}}ex_i) = ex_i \subseteq decod(A) \subseteq \psi_i$. Nach Voraussetzung folgt dann $\psi_{S(decod(A))} = \psi_i$ und damit $L_{\mathcal{L}S(A)} = L_i$.

„$\Leftarrow$": Gelte nun $\mathcal{L}_U \in$ CTG-FIN$_{\mathcal{L}_\psi}$. D.h. es existieren $({}_{\mathcal{L}}ex_i)_{i\geq 0}$ und $\mathcal{L}S$ mit den entsprechenden Eigenschaften. Setze wieder $ex_i = decod({}_{\mathcal{L}}ex_i)$ für alle i und $S(A) = \mathcal{L}S(cod(A))$. Damit ist $(ex_i)_{i\geq 0}$ r.g. und S berechenbar. Sei nun $\psi_i \in U$ beliebig und gelte $ex_i \subseteq A \subseteq \psi_i$. Es folgt $cod(ex_i) = {}_{\mathcal{L}}ex_i \subseteq cod(A) \subseteq L_i$. Die Voraussetzung impliziert $L_{\mathcal{L}S(cod(A))} = L_i$, also $\psi_{S(A)} = \psi_i$. $\square$

Das folgende Lemma beschreibt eine notwendige Bedingung an einen Hypothesenraum, wenn in ihm eine Klasse $\mathcal{L}$ gelernt werden soll.

2.11 Lemma. Wenn $\mathcal{L} \in$ TG-FIN$_{\mathcal{L}'}$ für ein $\mathcal{L} \in \mathcal{RA}$; $\mathcal{L}' = (L'_i)_{i\geq 0}$. Dann ist „$L'_i = \emptyset$?" uniform in i entscheidbar.

Beweis: Es gibt zwei Möglichkeiten: $\emptyset \notin \mathcal{L}$ oder $\emptyset \in \mathcal{L}$. Im ersten Fall entscheidet offensichtlich 0^∞ die Frage richtig. Beachte, daß $\mathcal{L}$ und $\mathcal{L}'$ die gleichen Sprachen enthalten müssen.
2. Fall. Angenommen, $\mathcal{L} \in$ TG-FIN bzgl. $\mathcal{L}'$. Dann existiert eine r.g. Familie $(ex_i)_{i\geq 0}$ guter Beispiele. Für alle i mit $L'_i = \emptyset$ muß $ex_i = \emptyset$ gelten, wegen $ex_i \subseteq L'_i$. Da $\emptyset \subseteq L'_j$ für alle j, muß also für alle L'_j mit $L'_j \neq \emptyset$ gelten: $ex_j \neq \emptyset$, sonst würde ein Widerspruch auf die übliche Art folgen: Die Strategie könnte nicht anhand $ex_i = ex_j$ zwischen den

unterschiedlichen Sprachen L'_i und L'_j unterscheiden.
Also gilt für alle i: $L'_i = \emptyset \iff ex_i = \emptyset$. Da $(ex_i)_{i\geq 0}$ uniform in i r.g., ist auch $ex_i = \emptyset$ uniform in i entscheidbar. $\square$

2.12 Satz. TG-FIN = TG-LIM $\subset$ CTG-FIN $\subset$ CTG-LIM $\subset \mathcal{RA}$

Beweis: Die Gleichheit gilt nach 2.2. TG-FIN $\subseteq$ CTG-FIN folgt sofort aus den entsprechenden Definitionen. Es bleibt die echte Inklusion zu zeigen.

Definiere dazu eine Sprachfamilie $\mathcal{L}_\Phi = (L'_i)_{i\geq 0}$ wie folgt. Φ ist ein Blum'sches Komplexitätsmaß. Für alle i sei

$$L'_i = \{y \mid \exists j, x \leq i[\Phi_j(x)\downarrow \text{ und } y \leq \Phi_j(x)]\}.$$

Offensichtlich ist $\mathcal{L}_\Phi$ uniform rekursiv aufzählbar.

Behauptung 1: $\mathcal{L}_\Phi \notin$ TG-FIN.

Beweis: Angenommen, $\mathcal{L}_\Phi \in$ TG-FIN bezüglich eines Hypothesenraumes $\mathcal{L}$ mit $\mathcal{L} = \mathcal{L}_\Phi$. Dann gibt es also gute Beispiele $(ex_i)_{i\geq 0}$ mit den gewünschten Eigenschaften. Für alle i, j gilt $L_i \subseteq L_j$ oder $L_j \subseteq L_i$. Sei weiterhin $k_i = max\, L'_i$ für alle i. Nach Definition von $\mathcal{L}_\Phi$ gilt $k_i \leq k_{i+1}$. Weiter ist die Folge der k_i natürlich unbeschränkt.

Betrachte nun zwei Sprachen L_i und L_j mit $L_i \subset L_j$. Dann muß $ex_j \setminus L_i \neq \emptyset$ gelten, weil sonst ein Widerspruch auf die übliche Art folgt. Aus der Konstruktion von $\mathcal{L}_\Phi$ folgt, daß der Fall des echten Enthaltenseins unendlich oft vorkommt. Setze $e_i = max\, ex_i$ für alle i und bilde nun daraus eine unendliche Teilfolge $m'_1, m'_2, \ldots$ mit $m'_i < m'_{i+1}$. Definiere nun weiter $m_i = m'_{i+1}$ für alle i. Dann ist diese Folge (m_i) wegen der rekursiven Erzeugbarkeit der guten Beispiele effektiv zu konstruieren. Es folgt $k_i < m_i$ für alle i, denn k_1 ist das „kleinstmögliche Maximum" der Rechenzeiten. Da m_1 aus einer echten Differenz stammt, muß es also größer sein. Der Rest folgt, weil die k_i „so langsam wie möglich" wachsen, während in $\mathcal{L}$ – bzw. in den guten Beispielen – höchstens größere vorgezogen werden können, d.h. die Folge der m_i wächst mindestens so schnell wie die der k_i.

Aber $k_i = max\{k \mid$ es gibt $m, n \leq i$ so daß $\Phi_m(n)\downarrow$ und $\Phi_m(n) = k\}$, d.h. alle Funktionen φ_m, $0 \leq m \leq i$, sind auf den Argumenten $n \leq i$ entweder in höchstens k_i Schritten definiert oder $\varphi_m(n)\uparrow$. Da $k_i < m_i$ ist m_i eine effektiv berechenbare Rechenzeitschranke. Aber damit ist das Halteproblem offensichtlich entscheidbar: $\varphi_i(i)\downarrow \iff \Phi_i(i) \leq m_i$. $\lightning$ Die Behauptung folgt.

Behauptung 2: $\mathcal{L}_\Phi \in$ CTG-FIN.

Beweis: $\mathcal{L}_\Phi \subset$ FINITE $\in$ CTG-FIN. FINITE $\in$ TG-FIN ist offensichtlich; diese Familie ist in 1.7 so definiert worden, daß sie r.g. ist. Also können als gute Beispiele jeweils die ganze Sprache gewählt werden.

Das zeigt TG-FIN $\subset$ CTG-FIN.

Die zweite Inklusion folgt aus CTG-FIN $\subseteq$ CTG-LIM – siehe die entsprechenden Definitionen– und die echte Inklusion folgt aus Lemma 2.9.

Sei $\mathcal{L}_1$ stets die folgende indizierte Sprachfamilie.

$\mathcal{L}_1$	0	1	2	3	4
L_0	1	1	1	1	...
L_1	0	1	1	1	...
L_2	1	0	1	1	...
L_3	1	1	0	1	...
	...				

$\mathcal{L}_1$ liegt nicht in CTG-LIM: Jede Wahl von ex_0 führt dazu, daß $ex_0 \subseteq L_i$ für ein i mit $L_i \neq L_0$. Aber es gilt $ex_i \subseteq L_0$, da L_0 die Allsprache ist. Also muß sich jede Strategie auf einer der beiden Sprachen irren. Für einen *beliebigen* Hypothesenraum $\mathcal{L}'$ ist die Argumentation völlig analog; wähle „als L_0“ einfach ein L'_j, das die Allsprache ist. □

Könnte es nicht vielleicht für jede Klasse $\mathcal{L} \in$ CTG-LIM (bzw. CTG-FIN) eine Klasse $\mathcal{L}' \supseteq \mathcal{L}$ geben, mit $\mathcal{L}' \in$ TG-FIN? Das dem zumindest für CTG-LIM nicht so ist, belegt das folgende Korollar.

2.13 Korollar. a) Es gibt kein $\mathcal{L} \in \mathcal{RA}$, so daß $\mathcal{L} \supseteq \mathcal{L}_\mathcal{R}$ und $\mathcal{L} \in$ TG-FIN.

b) TG-FIN ist nicht $\subset$-abgeschlossen.

Beweis: Zu a): Angenommen, so ein $\mathcal{L}$ existiert. Dann folgt aus 2.12 sofort $\mathcal{L} \in$ CTG-FIN und damit $\mathcal{L}_\mathcal{R} \in$ CTG-FIN, im Widerspruch zu Lemma 2.9.
Zu b): Betrachte $\mathcal{L}_\Phi$ und FINITE. □

Für CTG-FIN ist bislang keine solche Klasse bekannt und auch die folgenden Beispiele geben keine endgültige Antwort auf diese Frage.

2.14 Definition: Nenne folgende, offensichtlich uniform r.a. Sprachfamilie $\mathcal{L}_K$ für ein beliebiges φ. Dabei bedeutet „$\varphi_0(0)\downarrow$?“, daß eine 1 steht, falls $\varphi_0(0)\downarrow$, 0 oder $\uparrow$ sonst.

$\mathcal{L}_K$	0	1	2	3	4
L_0	„$\varphi_0(0)\downarrow$?“	0	0	0	...
L_1	0	„$\varphi_1(1)\downarrow$?“	0	0	...
L_2	0	0	„$\varphi_2(2)\downarrow$?“	0	...
L_3	0	0	0	„$\varphi_3(3)\downarrow$?“	...
	...				

Definiere eine Klasse $\mathcal{L}_0$ wie folgt:

$\mathcal{L}_0$	0	1	2	3	4
L_0	0	0	0	0	...
L_1	1	0	0	0	...
L_2	0	1	0	0	...
L_3	0	0	1	0	...
L_4	0	0	0	1	...
	...				

Das folgende Beispiel ist nur auf den ersten Blick erstaunlich, sieht es doch so aus, als wäre damit das Halteproblem für φ entscheidbar: Die guten Beispiele sind r.g. und die Strategie arbeitet finit. In $\mathcal{L}_K$ gilt $L_i = \emptyset \iff \varphi_i(i)\uparrow$ und damit könnte mit Lemma 2.11 das Halteproblem entschieden werden.

2.15 Beispiel. $\mathcal{L}_K \in$ TG-FIN

Beweis: Der Trick ist, $\mathcal{L}_K$ bzgl. einer indizierten Familie $\mathcal{L}$ zu erlernen, die genau die gleichen Sprachen enthält. Definiere dazu $\mathcal{L} = (L_{\langle i,j\rangle})$ für alle i, j wie folgt.

$$\begin{aligned} L_{\langle 0,j\rangle} &= \emptyset \text{ für alle } j. \\ L_{\langle i+1,j\rangle} &= \begin{cases} \emptyset & \text{falls } \varphi_i(i) \text{ nicht in } \leq j \text{ Schritten definiert} \\ \{i\} & \text{sonst.} \end{cases} \end{aligned}$$

Damit ist $\mathcal{L}$ offensichtlich uniform rekursiv und enthält die gleichen Sprachen wie $\mathcal{L}_K$. Definiere nun die guten Beispiele für alle i und j:

$$\begin{aligned} ex_{\langle 0,j\rangle} &= \emptyset \text{ für alle } j. \\ ex_{\langle i+1,j\rangle} &= \begin{cases} \emptyset & \text{falls } i \notin L_{\langle i+1,j\rangle} \\ \{i\} & \text{sonst} \end{cases} \end{aligned}$$

Die guten Beispiele sind r.g. und für alle i, j gilt $ex_{\langle i,j\rangle} \subseteq L_{\langle i,j\rangle}$. Beachte, daß es hier keine „störenden“ Obermengen A geben kann. Die Strategie S ist nun sehr einfach. Sei A eine endliche Menge von Worten.

$$S(A) = \begin{cases} \langle 0,0\rangle & \text{falls } A = \emptyset \\ \mu j[i \in L_{\langle i+1,j\rangle}] & \text{falls } A = \{i\} \\ \uparrow & \text{sonst} \end{cases}$$

Augenscheinlich ist S berechenbar. Sei nun $\langle i,j\rangle$ beliebig. Falls $ex_{\langle i,j\rangle} = \emptyset$, dann $L_{\langle i,j\rangle} = \emptyset$ und $S(ex_{\langle i,j\rangle}) = \langle 0,0\rangle$. Weil $L_{\langle 0,0\rangle} = \emptyset$, erlernt S diese Sprache. Falls $ex_{\langle i,j\rangle} \neq \emptyset$, dann $L_{\langle i,j\rangle} = ex_{\langle i,j\rangle}$. Aber dann $S(ex_{\langle i,j\rangle}) = \langle i,j'\rangle$ mit $\{i-1\} \subseteq L_{\langle i,j'\rangle}$. (Beachte, daß $i > 0$ wegen $ex_{\langle i,j\rangle} \neq \emptyset$ und $L_{\langle 0,j\rangle} = \emptyset$ für alle j.) Aber dann folgt aus der Definition von $\mathcal{L}$ sofort $L_{\langle i,j\rangle} = L_{\langle i,j'\rangle}$, und S lernt auch in diesem Fall die richtige Sprache. □

$\mathcal{L}$ erfüllt natürlich auch die notwendige Bedingung aus Lemma 2.11. Das ist aber kein Widerspruch, denn um das Halteproblem zu entscheiden, müßte es bekannt sein, für welches $\langle i,j\rangle$ nun „$ex_{\langle i,j\rangle} = \emptyset$?“ zu testen ist. Es folgt allerdings, daß $\mathcal{L}_K$ unmöglich bezüglich $\mathcal{L}_K$ zu erlernen ist.

Wenn es nun für jedes $\mathcal{L} \in \mathcal{RA}$ mit $\mathcal{L} \in$ TG-FIN eine indizierte Familie $\mathcal{L}'$ mit den gleichen Sprachen gibt, für die $\mathcal{L}' \in$ TG-FIN gilt, dann wäre die ganze Betrachtung mehr oder weniger sinnlos. Es würde in dem Fall genügen, sich auf das Studium uniform rekursiver Sprachfamilien zu beschränken. Das nächste Beispiel gibt eine Sprachfamilie an, für die es keine entsprechende indizierte Familie gibt.

2.16 Beispiel. Sei φ eine Numerierung von $\mathcal{P}$. Definiere $\mathcal{L}_2 = (L2_i)_{i\geq 0}$ durch

$$L2_i = \begin{cases} \{2i, 2i+1\} & \text{falls } i \in K \\ \{2i\} & \text{falls } i \notin K \end{cases}$$

$\mathcal{L}_2$ ist offensichtlich uniform r.a. und jede Zahl kann in höchstens einer Sprache vorkommen.
Behauptung: Es gibt keine uniform rekursive Sprachfamilie $\mathcal{L}$, die genau die gleichen Sprachen enthält.
Beweis: Angenommen, so ein $\mathcal{L}$ exsistiert und $\chi_{\mathcal{L}}$ entscheidet uniform das Wortproblem für $\mathcal{L}$. Definiere dann

$\chi_K(i) =$ „Suche j mit $2i \in L_j$. Falls $2i+1 \in L_j$ gib „1“ aus, sonst gib „0“ aus. Stop.“

Offensichtlich ist χ_K eine rekursive Funktion und entscheidet nach Definition von $\mathcal{L}_2$ das Halteproblem. ↯

Behauptung: $\mathcal{L}_2 \in$ TG-FIN.
Beweis: Als Hypothesenraum wird $\mathcal{L}_2$ gewählt. Setze $ex_i = \{2i\}$ für alle i. Damit ist $(ex_i)_{i\geq 0}$ r.g. Die Strategie ist ebenfalls sehr einfach: $S(A)$ sucht in A nach einem Wort der Gestalt $2i$ und gibt i zurück. ◇

2.17 Beispiel. Definiere $\mathcal{K} = (K_i)$, wobei $K_i = \{j \mid j \leq i \text{ und } \varphi_j(j)\downarrow\}$ für alle i. FINITE $= (E_i)_{i\geq 0}$ ist eine injektive, r.g. Aufzählung aller endlichen Mengen.

a) Es gilt $\mathcal{K} \in$ TG-FIN. Allerdings kann $\mathcal{K}$ nicht bzgl. des Hypothesenraumes $\mathcal{K}$ gelernt werden, denn dann wäre das Halteproblem entscheidbar.

Definiere nun einen Hypothesenraum $\mathcal{L} = (L_i)_{i\geq 0}$ wie folgt. Sei dazu c eine Funktion, die K ohne Wiederholungen aufzählt; $c_1, c_2, \ldots$. Setze nun $L_0 = \emptyset$ und für alle $i \geq 1$: $L_i = \{j \mid j \leq c_i \text{ und } \varphi_j(j)\downarrow\}$. Dann ist $\mathcal{L}$ offensichtlich uniform r.a. Weiter ist c_i das maximale Wort in L_i und effektiv berechenbar.

Die guten Beispiele werden für alle $i \geq 1$ durch $ex_i = \{c_i\}$ definiert ($ex_0 = \emptyset$) und sind damit r.g. Sei A eine beliebige endliche Menge von Worten. Dann ist

$$S(A) = \mu k[max(ex_k) = max(A)]$$

sicher berechenbar und erlernt $\mathcal{K}$ bzgl. $\mathcal{L}$ im TG-FIN-Sinn.

b) Definiere für alle i: $L_i = E_i \cap K_i$. Offensichtlich ist $\mathcal{L} = (L_i)_{i\geq 0}$ aus $\mathcal{RA}$. Der einfache Trick aus a) geht nun nicht mehr, weil in der Regel nicht mehr bekannt ist, mit welchem E_i die Menge $\{j \mid j \leq c_i \text{ und } \varphi_j(j)\downarrow\}$ geschnitten werden müsste. Allerdings gibt es ein $\mathcal{L}' \in \mathcal{IF}$ mit $\mathcal{L} = \mathcal{L}'$, in dem $\mathcal{L}$ gelernt werden kann. Der Beweis soll hier nur informell angegeben werden. Sei $c_1, c_2, \ldots$ wieder eine injektive Aufzählung von K. Bilde $\mathcal{L}' = (L'_i)_{i\geq 0}$ nun dadurch, daß $\mathcal{L}'$ zuerst alle Teilmengen von $\emptyset$ aufzählt, dann die von $\{c_1\}$, $\{c_1, c_2\}$, $\{c_1, c_2, c_3\}$, usw. Dann ist $\mathcal{L}'$ sogar r.g. Ein kurzer Vergleich mit $\mathcal{L}$ ergibt $\mathcal{L} = \mathcal{L}'$. Definiere $ex'_i = L'_i$ für alle i und $S(A)$ ist einfach die kleinste Nummer der Sprache A in $\mathcal{L}'$. Dann folgt $\mathcal{L} \in$ TG-FIN bzgl. $\mathcal{L}'$. ◇

2.18 Beispiel. Definiere für alle i: $L_i = \{\langle i, j\rangle \mid \varphi_i(j)\downarrow\}$ und setze $\mathcal{L} = (L_i)_{i\geq 0}$. Dann gilt $\mathcal{L} \in$ TG-FIN.

Der Beweis ist sehr einfach. Wähle als Hypothesenraum eine injektive Aufzählung $\mathcal{L}' = (L'_i)_{i\geq 0}$ von $\mathcal{L}$, die $L'_0 = \emptyset$ erfüllt. $\mathcal{L}'$ kann mit Standardkonstruktionen aus $\mathcal{L}$ gewonnen werden. Definiere $ex_0 = \emptyset$ und für alle $i \geq 1$:

$$ex_i = \text{„Suche ein } x \in L'_i \text{ und gibt } \{x\} \text{ zurück.“}$$

Offensichtlich ist $(ex_i)_{i\geq 0}$ r.g. Die Strategie sucht eine konsistente Hypothese, gibt sie aus und stoppt. $\diamond$

$\mathcal{L}$ enthält quasi jede r.a. Menge – dargestellt durch *domain* φ_i, zusammen mit „ihrer Nummer" i. Beachte, daß $\mathcal{W} = (W_i)_{i\geq 0}$, $W_i = \{j \mid \varphi_i(j)\downarrow\}$ allerdings nicht aus TG-FIN ist, denn $\mathcal{W}$ enthält alle endlichen Sprachen und und mindestens eine unendliche.

In [LNW] wurde folgendes Lemma für das Lernen von indizierten Sprachfamilien bewiesen.

2.19 Lemma. Sei $\mathcal{L}' \in$ TG-FIN$_{\mathcal{L}}$. Dann ist die Gleichheit in $\mathcal{L}$ entscheidbar. (D.h. es gibt eine rekursive Funktion $\chi_=$ mit $\chi_=(i,j) = 1 \iff L_i = L_j$.)

Diese Aussage läßt sich dahingehend verschärfen, daß immer eine geeignete injektive Aufzählung als Hypothesenraum möglich ist. Mit anderen Worten: Jedes TG-FIN-lösbare Lernproblem indizierter Familien läßt sich in einem „minimalen" Hypothesenraum lösen.

2.20 Korollar. Sei $\mathcal{L}' = (L'_i)_{i\geq 0} \in \mathcal{IF}$ und gelte $\mathcal{L}' \in$ TG-FIN. Dann gibt es ein injektives $\mathcal{L} \in \mathcal{IF}$ mit $\mathcal{L}' = \mathcal{L}$ und $\mathcal{L}' \in$ TG-FIN$_{\mathcal{L}}$.

Beweis: Nach Voraussetzung gilt $\mathcal{L}' \in$ TG-FIN$_{\mathcal{L}''}$ für eine geeignete indizierte Familie $\mathcal{L}''$, mit guten Beispielen $(ex''_i)_{i\geq 0}$. Nach 2.19 ist dann in $\mathcal{L}''$ die Gleichheit entscheidbar. Daraus wird nun der gesuchte Hypothesenraum $\mathcal{L}$ gewonnen, indem alle Duplikate aus $\mathcal{L}''$ gestrichen werden. Das ergibt unmittelbar einen rekursiven Compiler $c : \mathcal{L} \mapsto \mathcal{L}''$. Definiere nun $ex_i = ex''_{c(i)}$ für alle $i \geq 0$. Offensichtlich ist $(ex_i)_{i\geq 0}$ r.g. und erfüllt die Bedingungen in Satz 2.1, weil $(ex''_i)_{i\geq 0}$ sie erfüllt. Die Behauptung folgt. $\square$

Beim Lernen rekursiver Funktionen gilt ein Analogon zu 2.19. Hier ist allerdings zu beachten, daß Gex-Fin eher CTG-FIN entspricht, d.h. der Hypothesenraum kann auch nicht zu erlernende Funktionen enthalten.

2.21 Lemma. Sei $U \subseteq \mathcal{R}$. Wenn $U \in$ Gex-Fin$_\psi$, dann existiert eine partiell rekursive Funktion G, so daß für alle i, j mit $\psi_i, \psi_j \in U$ gilt

$$G(i,j) = \begin{cases} 1 & \text{falls } \psi_i = \psi_j \\ 0 & \text{sonst.} \end{cases}$$

Anschaulich besagt 2.21, daß für den U entsprechenden Teil von ψ die Funktionengleichheit entscheidbar ist.

Beweis: Der Beweis ähnelt sehr dem von Satz 2.1, nur $\psi_i \notin U$ muß gesondert betrachtet werden. Sei also $U \in$ Gex-Fin$_\psi$.

Definiere $G(i,j) =$ „ Berechne ex_i, ex_j. Teste $ex_i \subseteq \psi_j$ und $ex_j \subseteq \psi_i$. Falls einer der beiden Schritte nicht terminiert, dann ist $G(i,j)$ ebenfalls undefiniert. Falls der Test erfolgreich endet, dann gib 1 aus; Stop. Falls der Test ohne Erfolg endet, gib 0 aus; Stop."

Dann ist G berechenbar. Für $\psi_i, \psi_j \in U$ ist die Konstruktion von ex_i, ex_j endlich und der darauffolgende Test effektiv, weil ψ_i, ψ_j total sind. Falls der Test ohne Erfolg bleibt, dann sind ψ_i, ψ_j offensichtlich verschiedene Funktionen. Ist der Test hingegen erfolgreich, dann gilt $\psi_i = \psi_j$, denn ansonsten könnte die nach Voraussetzung existierende Strategie bei der Eingabe $ex_i \cup ex_j \subseteq L_i \cap L_j$ nicht zwischen diesen ungleichen Funktionen unterscheiden.

Für $\psi_i \notin U$ ist weder klar, ob ex_i endlich erzeugbar, noch ob ψ_i total ist. Dementsprechend können die Konstruktionen oder Tests in der Definition von G eventuell nicht termineren oder falsche Ergebnisse liefern, und in diesen Fällen ist G dann undefiniert oder falsch. $\Box$

Im Unterschied dazu gibt es uniform rekursiv aufzählbare Sprachfamilien, die zwar aus TG-FIN sind, für die aber in keinem Hypothesenraum die Gleichheit entscheidbar ist. Diese Aussage steht der in Korollar 2.20 fast diametral gegenüber. Zudem legt sie zwei Vermutungen nahe: 1.) Das Lernen von indizierten Familien und uniform r.a. Familien ähneln sich vielleicht nicht so sehr, wie es Satz 2.1 suggeriert. 2.) Die guten Beispiele einer Sprachklasse nach 2.22 geben keinen Aufschluß über die Ähnlichkeit ihrer Sprachen. Hätten etwa gleiche Sprachen gleiche Beispiele, dann wäre die Gleichheit in $\mathcal{L}$ natürlich entscheidbar. Weil $\mathcal{L}$ trotzdem gelernt werden kann, müssen die Beispiele Information tragen. Aber die scheint mehr auf der Numerierung als auf der Sprache selbst zu beruhen.

2.22 Satz. Es gibt ein $\mathcal{L} \in \mathcal{RA}$, für das gilt

- $\mathcal{L} \in$ TG-FIN,
- es gibt keine injektive Aufzählung von $\mathcal{L}$.

Insbesondere kann es dann keine Aufzählung von $\mathcal{L}$ geben, in der die Gleichheit entscheidbar ist, denn daraus würde die Existenz einer injektiven Aufzählung folgen. Wenn also $\mathcal{L} \in$ TG-FIN bzgl. eines Hypothesenraumes $\mathcal{L}'$, dann kann die Gleicheit in $\mathcal{L}'$ nicht entscheidbar sein.

Beweis: Das Alphabet sei $\{a, b\}$. Definiere $\mathcal{L} = (L_{i,j})_{i \geq 0, j \in \{1,2\}}$ für alle $i \in I\!N$ wie folgt.

$$\begin{aligned} L_{i,1} &= \{a^i, a^i b^i\} \\ L_{i,2} &= \begin{cases} \{a^i, a^i b^i\} & \text{falls } \varphi_i(i)\downarrow \\ \{a^i\} & \text{sonst.} \end{cases} \end{aligned}$$

Definiere nun die guten Beispiele für alle i durch

$$ex_{i,j} = \begin{cases} \{a^i, a^i b^i\} & \text{falls } j = 1 \\ \{a^i\} & \text{falls } j = 2. \end{cases}$$

Offensichtlich ist diese Mengenfamilie r.g. Die folgende Strategie ist berechenbar für jede endliche Menge A:

$$S(A) = \begin{cases} i, 1 & \text{falls } a^i, a^i b^i \in A \\ i, 2 & \text{falls } a^i \in A \text{ und } a^i b^i \notin A. \end{cases}$$

(In zweideutigen Fällen ist S undefiniert.) Sei nun $L_{i,j}$ zu erlernen und gelte $ex_{i,j} \subseteq A \subseteq L_{i,j}$. Falls $\varphi_i(i)\downarrow$, dann $L_{i,1} = L_{i,2}$ und S liegt auf jeden Fall richtig. Falls $\varphi_i(i)\uparrow$, dann $a^i b^i \in L_{i,j} \iff j = 1$ und wieder gibt S die richtige Antwort. Also $\mathcal{L} \in$ TG-FIN bzgl. $\mathcal{L}$.

Es bleibt zu zeigen, daß keine injektive Numerierung von $\mathcal{L}$ existiert. Es gilt augenscheinlich $L_{i,1} \neq L_{i,2} \iff \varphi_i(i)\uparrow$. Angenommen, $\mathcal{L}'$ sei eine injektive Numerierung von $\mathcal{L}$. Dann gibt es für alle i genau dann zwei Sprachen in $\mathcal{L}'$, die a^i enthalten, wenn $\varphi_i(i)\uparrow$. Damit ließe sich nun $\overline{K}$ aufzählen: „Gib alle i aus, für die es zwei Sprachen in $\mathcal{L}'$ gibt, die a^i enthalten." Also gibt es keine injektive Aufzählung von $\mathcal{L}$. □

Auch die folgende, aus der Literatur – siehe [PP] – bekannte Sprachklasse kann zum Beweis des letzten Satzes verwendet werden.

$$\mathcal{L} = \{\{2i\}, \{2i+1\} \mid i \notin K\} \cup \{\{2i, 2i+1\} \mid i \in K\}$$

Diese Klasse besitzt ebenfalls keine injektive Numerierung und ist offensichtlich aus TG-FIN.

Nun zu einer Art von uniform r.a. Sprachfamilien, die anscheinend in der Literatur recht häufig vorkommen. Zum Beispiel für eine beliebige Numerierung φ sei $\mathcal{L}_\varphi$ wie folgt definiert:

$\mathcal{L}_\varphi$	0	1	2	3	4
L_0	„$\varphi_0(0)\downarrow$?"	0	0	0	...
L_1	1	„$\varphi_1(1)\downarrow$?"	0	0	...
L_2	1	1	„$\varphi_2(2)\downarrow$?"	0	...
L_3	1	1	1	„$\varphi_3(3)\downarrow$?"	...
	...				

Bezeichnung: Ein $\mathcal{L} \in \mathcal{RA}$ heißt *Anfangsstücksammlung*, wenn für jede Sprache L_i aus $\mathcal{L}$ gilt $L_i = 1^{f(i)}0^\infty$. Dabei muß f nicht unbedingt berechenbar sein.

Offensichtlich ist $\mathcal{L}_\varphi$ eine Anfangsstücksammlung.

2.23 Lemma. Für eine Anfangsstücksammlung $\mathcal{A}$ gilt $\mathcal{A} \in$ TG-FIN $\iff$ Es gibt ein $\mathcal{L} \in \mathcal{RA}$ mit $\mathcal{A} = \mathcal{L}$ und eine r.g. Familie $(ex_i)_{i\geq 0}$, so daß für alle i, j gilt:

1. $ex_i \subseteq L_i$,
2. $ex_i \subseteq L_j \Longrightarrow L_i \subseteq L_j$.

Diese Bedingung ähnelt sehr der Charakterisierung von Lange und Zeugmann für T-FIN für indizierte Sprachklassen, siehe [LZ a]. Es folgt der sehr einfache

Beweis: Beachte, daß für alle i, j entweder $L_i \subset L_j$ oder $L_j \subset L_i$ oder $L_i = L_j$ gilt. $(\otimes)$

„ $\Leftarrow$ " : Definiere $S(A) =$ „Ein k mit $ex_k \subseteq A \subseteq L_k$." für alle endlichen Mengen A. Seien nun i, A mit $ex_i \subseteq A \subseteq L_i$ beliebig. Dann ist $S(A)\downarrow$, etwa $S(A) = k$. Also gilt $ex_k \subseteq A \subseteq L_k$ und damit $ex_k \subseteq L_i$ und $ex_i \subseteq L_k$, woraus mit 2.1 folgt $L_i = L_k$.

„ $\Rightarrow$ " : Wenn $A \in$ TG-FIN bzgl. $\mathcal{L}$, dann erfüllen die guten Beispiele die Bedingungen aus Satz 2.1. Damit folgt die Behauptung sofort aus $\otimes$. □

Was beim Beweis eigentlich verwendet wurde, war $\otimes$. Das Lemma kann deshalb auch allgemeiner formuliert werden.

2.24 Satz. Sei $\mathcal{L} = (L_i)_{i \geq 0} \in \mathcal{RA}$, so daß für alle i und j entweder $L_i = L_j$ oder $L_i \subset L_j$ oder $L_j \subset L_i$ gilt. Dann gilt $\mathcal{L} \in$ TG-FIN $\iff$ Es gibt ein $\mathcal{L}' \in \mathcal{RA}$ mit $\mathcal{L} = \mathcal{L}' = (L'_i)_{i \geq 0}$ und eine r.g. Familie $(ex_i)_{i \geq 0}$, so daß für alle i, j gilt:

1. $ex_i \subseteq L'_i$,
2. $ex_i \subseteq L'_j \Longrightarrow L'_i \subseteq L'_j$.

Der Satz ist alles andere als tiefschürfend. Anschaulich besagt er, daß die guten Beispiele für eine Sprache wenigstens ein Wort enthalten müssen, daß in keiner echten Teilsprache der zu erlernenden Sprache vorkommt. Er ist hier hauptsächlich deswegen aufgeführt, weil er den formalen Beweis erbringt, daß „Occam's Razor" im betrachteten Modell „gilt".

Nun zu einer Tatsache, die mich doch erstaunt hat. Normalerweise werden Gödelnumerierungen immer in der Form „sei φ eine beliebige Gödelnumerierung" verwendet, d.h. es ist egal, welche Gödelnumerierung ausgewählt wird. Der folgende Satz zeigt, daß das Halteproblem für verschiedene Gödelnumerierungen – in dieser Repräsentation – verschieden schwer zu lernen ist.

2.25 Satz. Es gibt Gödelnumerierungen φ, ψ mit

1. $\mathcal{L}_\varphi \in \text{TG-FIN}_{\mathcal{L}_\varphi}$,
2. $\mathcal{L}_\psi \notin \text{TG-FIN}_{\mathcal{L}_\psi}$.

Beweis: Sei ζ eine beliebige Gödelnumerierung. Definiere nun für alle i

$$\varphi_i = \begin{cases} \zeta_j & i = 2j \\ \emptyset & i = 2j+1 \end{cases}$$

und

$$\psi_i = \begin{cases} \zeta_j & i = 2j \\ f_j & i = 2j+1, \end{cases}$$

wobei für alle j und x

$$f_j(x) = \begin{cases} 2 & x = 0 \\ 0 & x > 0 \text{ und } \zeta_j(2j+1) = 0 \\ \uparrow & \text{sonst.} \end{cases}$$

Offensichtlich sind φ und ψ Gödelnumerierungen. Für beliebige Numerierungen ψ von $\mathcal{P}$ gilt das folgende Lemma, aus dem sich die Aussage des Satzes fast unmittelbar ergibt. Beachte, daß nach Definition von $\mathcal{L}_\psi$ dann auch $max\ L_i \leq i$ gilt. Sei K_ψ das Halteproblem für die Numerierung ψ.

Lemma: Sei ψ eine beliebige Numerierung von $\mathcal{P}$. $\mathcal{L}_\psi \in$ TG-FIN$_{\mathcal{L}_\psi}$ $\iff$ Es gibt eine rekursive Menge $E \subseteq K_\psi$, so daß für alle $i, i+1 \in K_\psi$ gilt: $i+1 \in E$.

Bew.: „ $\Rightarrow$ “ : Gelte also $\mathcal{L}_\psi \in$ TG-FIN$_{\mathcal{L}_\psi}$ und seien $(ex_i)_{i \geq 0}$ die guten Beispiele. Definiere $E = \{i \mid \mathit{max}\, ex_i = i\}$. Dann ist E rekursiv und eine Teilmenge von K_ψ. Wenn E nun nicht die Forderung auf der rechten Seite erfüllt, dann gibt es $i, i+1 \in K_\psi$ mit $\mathit{max}\, ex_{i+1} < i+1$. Damit gilt $ex_i \subseteq L_{i+1}$, $ex_{i+1} \subseteq L_i$ und $L_i \neq L_{i+1}$. Aber das ist mit Satz 2.1 ein Widerspruch zur Voraussetzung.

„ $\Leftarrow$ “ : Definiere für alle i

$$ex_i = \begin{cases} \{0, \ldots, i\} & i \in E \\ \{0, \ldots, i-1\} & i \notin E. \end{cases}$$

Dann ist $(ex_i)_{i \geq 0}$ r.g. und es gilt $ex_i \subseteq L_i$ für alle i. Definiere für alle endlichen Mengen A:

$$S(A) = \text{„Suche ein } i \text{ mit } ex_i \subseteq A \subseteq L_i\text{.“}$$

Seien also i, A mit $ex_i \subseteq A \subseteq L_i$ beliebig. Dann ist $S(A)$ auf jeden Fall definert, etwa $S(A) = j$. Aus der Definition von $\mathcal{L}_\psi$ und den guten Beispielen folgt unmittelbar $i - 1 \leq j \leq i + 1$. Es sind nun verschiedene Fälle möglich.

1. $\underline{i \in K_\psi}$: Dann gilt $L_{i-1} \subset L_i$.
   (a) $\underline{i - 1 \in K_\psi}$: Dann folgt $i \in E$ und $ex_i = A = L_i$ und $A \not\subseteq L_{i-1}$, d.h. $j \geq i$.
      i. $\underline{i + 1 \notin K_\psi}$: Dann $L_{i+1} = L_i$ und S liegt also richtig.
      ii. $\underline{i + 1 \in K_\psi}$: Dann $i + 1 \in E$ und $ex_{i+1} = L_{i+1}$ und $ex_{i+1} \not\subseteq A$, d.h. $j \leq i$ und damit insgesamt $j = i$.
   (b) $\underline{i - 1 \notin K_\psi}$: Dann folgt $L_{i-1} = \{0, \ldots, i-2\}$ und $A \not\subseteq L_{i-1}$, d.h. $j \geq i$. Die Fälle für $i + 1$ sind völlig analog zu den Fällen bei (a).
2. $\underline{i \notin K_\psi}$: Dann $L_i = \{0, \ldots, i-1\}$ und $ex_i = A = L_i$. Weil $ex_{i+1} \supseteq \{0, \ldots, i\}$ folgt $ex_{i+1} \not\subseteq A$ und damit $j \leq i$.
   (a) $\underline{i - 1 \notin K_\psi}$: Dann $i - 1 \notin L_{i-1}$ und $A \not\subseteq L_{i-1}$, also $j \geq i$ und damit $j = i$.
   (b) $\underline{i - 1 \in K_\psi}$: Dann $L_{i-1} = L_i$ und S liegt wegen $j \leq i$ richtig.

Die Behauptung folgt. ∘

Daraus folgt unmittelbar $\mathcal{L}_\varphi \in$ TG-FIN$_{\mathcal{L}_\varphi}$, denn die rechte Seite des Lemmas ist offensichtlich erfüllt: Es gibt kein i mit $i \in K_\varphi$ und $i + 1 \in K_\varphi$.

Bleibt $\mathcal{L}_\psi \notin$ TG-FIN$_{\mathcal{L}_\psi}$ zu zeigen. Nach dem Lemma genügt es zu beweisen, daß keine wie im Lemma beschriebene Menge E existiert.
Angenommen, es gibt ein solches E für ψ. Sei ψ_k die charakteristische Funktion von E. Dann gilt $k = 2n$, denn $\mathit{range}\, \psi_k \subseteq \{0, 1\}$ und alle „ungeraden“ Funktionen in ψ –

also die f_j – haben eine 2 im Bildbereich. Betrachte nun ψ_{2n} und ψ_{2n+1}. Offensichtlich gilt $\psi_{2n}(2n)\downarrow$ und $\psi_{2n}(2n+1)\downarrow$, d.h. $2n \in K_\psi$. Weil ψ_{2n} ein rekursives Prädikat ist, sind genau zwei Fälle möglich.

$\psi_{2n}(2n+1) = 0:$ Dann $\psi_{2n+1} = f_n = 2\,0^\infty$, also $2n{+}1 \in K_\psi$. Damit gilt $\{2n, 2n{+}1\} \subseteq K_\psi$. Nach Lemma muß dann $2n+1 \in E$ gelten, aber $\psi_{2n}(2n+1) = 0$. $\lightning$

$\psi_{2n}(2n+1) = 1:$ Dann $\psi_{2n+1} = f_n = 2\uparrow^\infty$, also $2n+1 \notin K_\psi$. Weil $\psi_{2n}(2n+1) = 1$ gilt, folgt $E \not\subseteq K_\psi$. $\lightning$

Also existiert E nicht und die Behauptung folgt. $\square$

Auf ähnliche Weise lassen sich weitere Fakten über Gödelnumerierungen herleiten. Z.B. gibt es Gödelnumerierungen, für die es ein rekursives $E \subseteq K$ mit $x \in K \setminus E \to x{+}1 \in E$ für alle x gibt.

Kapitel 3

Textbeispiele und rekursive Strategien

Die in den Beweisen verwendeten Strategien sind normalerweise „Aufzählungsstrategien“: Sie durchsuchen einen Hypothesenraum nach konsistenten Hypothesen. Formal wird das durch eine unbeschränkte Minimierung ausgedrückt. Wenn es für eine Menge von Beispielen keine „passende“ Hypothese gibt divergiert eine solche Strategie. Im folgenden werden nun Strategien betrachtet, die für alle Eingaben definiert sind; d.h. sie müssen erkennen, wenn eine Eingabe keine Hypothese beschreibt, um stoppen zu können. Dieses Verhalten könnte so interpretiert werden, daß die Strategie ihre Kompetenz – sprich ihren Hypothesenraum – abschätzen kann.

3.1 Definition. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L} = (L_i)_{i \geq 0}$ eine Aufzählung von $\mathcal{L}'$.

$\mathcal{L}'$ heißt *anhand guter Textbeispiele finit bzgl. $\mathcal{L}$ mit kompetenter Strategie erlernbar*, wenn es $S \in \mathcal{R}$ und eine r.g. Familie $(ex_i)_{i \in I\!N}$ gibt, so daß für alle i gilt:

1. $ex_i \subseteq L_i$,
2. für alle $A \subseteq L_i$ existiert ein j mit $S(ex_i \cup A) = j$ und $L_j = L_i$,
3. für alle B gilt: Wenn es kein i mit $ex_i \subseteq B \subseteq L_i$ gibt, dann ist $S(B) =$ „Es gibt keine vernünftige Hypothese“.

Dies wird kurz $\mathcal{L}' \in$ TG-FIN-K$_{\mathcal{L}}$ geschrieben.

Der entsprechende Inferenztyp wird mit TG-FIN-K bezeichnet. •

Das ist sicher die schärfstmögliche Forderung an die Kompetenz einer Strategie: Wenn die Beispiele B gültig sind (d.h. wenn es eine Sprache i mit $ex_i \subseteq B \subseteq L_i$ gibt), dann muß sie richtig lernen; wenn sie nicht gültig sind, muß sie das erkennen. Etwas weicher wäre die Forderung nach einer „nur“ rekursiven Strategie. Und zwar weil die Strategie hier nicht mehr die „Ungültigkeit“ von Beispielen erkennen muß, sie könnte z.B. auf ungültigen Beispielen eine Hypothese ausgeben, die mit den Beispielen nichts zu tun hat. Es wäre dann die Aufgabe des Benutzers, die eventuelle Fehlerhaftigkeit der Hypothese zu erkennen. Aber immerhin hat er dazu die Möglichkeit, er muß nicht ewig auf eine Antwort von S warten. Wenn die Hypothese nicht seinen Erwartungen entspricht, kann er die Strategie mit neuen Beispielen füttern. Also:

3.2 Definition. Sei $\mathcal{L} \in \mathcal{RA}$. $\mathcal{L}'$ heißt *anhand guter Textbeispiele finit bzgl.* $\mathcal{L}$ *mit rekursiver Strategie erlernbar*, wenn $\mathcal{L}' \in$ TG-FIN$_{\mathcal{L}}$ und die verwendete Strategie rekursiv ist.

Der entsprechende Inferenztyp wird mit TG-FIN-WK bezeichnet. •

WK soll auf eine „weichere" Form der Kompetenz hinweisen. Die Ergebnisse sind in den folgenden zwei Sätzen zusammengefaßt. Vorher eine offensichtliche, aber nützliche Folgerung aus Definition 3.1.

3.3 Lemma. Sei $\mathcal{L}' \in \mathcal{RA}$. $\mathcal{L}' \in$ TG-FIN-K $\iff$ Es gibt ein $\mathcal{L} = (L_i)_{i \geq 0} \in \mathcal{RA}$ mit $\mathcal{L} = \mathcal{L}'$, eine r.g. Familie $(ex_i)_{i \geq 0}$ und ein Prädikat $d \in \mathcal{R}$, so daß für alle i, j und alle endlichen Mengen A gilt:

- $ex_i \subseteq L_i$,
- $(ex_i \subseteq L_j \wedge ex_j \subseteq L_i) \;\Rightarrow\; L_i = L_j$,
- $d(A) = 1$ genau dann, wenn $\exists k : ex_k \subseteq A \subseteq L_k$.

Beweis: „ $\Rightarrow$ ": Definiere

$$d(A) = \begin{cases} 1 & \text{falls } S(A) \in I\!N, \\ 0 & \text{falls } S(A) = \text{„Es gibt keine vernünftige Hypothese"}. \end{cases}$$

Weil S nach Voraussetzung rekursiv und richtig ist, folgen sofort die gewünschten Eigenschaften von d. Die Eigenschaften der guten Beispiele folgen aus 2.1, denn es gilt natürlich sofort nach Definition TG-FIN-K $\subseteq$ TG-FIN.

„ $\Leftarrow$ ": Gebe es also $\mathcal{L}$, $(ex_i)_{i \geq 0}$ und d wie gefordert. Definiere

$$S(A) = \begin{cases} \text{ein } k \text{ mit } ex_k \subseteq A \subseteq L_k & \text{falls } d(A) = 1, \\ \text{„Es gibt keine vernünftige Hypothese"} & \text{falls } d(A) = 0. \end{cases}$$

S hat offensichtlich die Eigenschaften aus Definition 3.1. □

3.4 Satz. Für indizierte Familien gilt TG-FIN-K = TG-FIN-WK $\subset$ TG-FIN.

Beweis: Zunächst der Beweis der Gleichheit. „ $\subseteq$ " folgt sofort aus den Definitionen.

„ $\supseteq$ ": Sei also $\mathcal{L}' \in$ TG-FIN-WK$_{\mathcal{L}''}$ mit der Strategie $S_{\mathcal{L}''} \in \mathcal{R}$ und den guten Beispielen $(ex''_i)_{i \geq 0}$. Nach 2.19 ist dann die Gleichheit in $\mathcal{L}''$ entscheidbar. Konstruiere aus $\mathcal{L}''$ eine injektive Aufzählung $\mathcal{L} = (L_i)_{i \geq 0}$ wie im Beweis von 2.20. Dann gibt es rekursive Compiler $c : \mathcal{L} \mapsto \mathcal{L}''$ und $c^{-1} : \mathcal{L}'' \mapsto \mathcal{L}$. Definiere $ex_i = ex''_{c(i)}$ für alle i und $S(A) = c^{-1}(S_{\mathcal{L}''}(A))$ für alle A. Damit ist $(ex_i)_{i \geq 0}$ r.g. und S rekursiv.

Seien i, A mit $ex_i \subseteq A \subseteq L_i$ beliebig. Dann $ex''_{c(i)} \subseteq A \subseteq L''_{c(i)}$. Weil $L''_{S_{\mathcal{L}''}(A)} = L''_{c(i)}$ folgt $L_{S(A)} = L_{c^{-1}(S_{\mathcal{L}''}(A))} = L''_{S_{\mathcal{L}''}(A)} = L''_{c(i)} = L_i$. Also kann auch in TG-FIN-WK immer bezüglich injektiven Hypothesenräumen gelernt werden.

Sei deshalb $\mathcal{L}''$ o.B.d.A. injektiv. Definiere nun eine Strategie S' für alle A wie folgt.

$S'(A) =$ „Berechne $j = S_{\mathcal{L}''}(A)$.
Wenn $ex''_j \subseteq A \subseteq L''_j$, dann gibt j aus. Stop.
Gib „Es gibt keine vernünftige Hypothese“ aus. Stop.“

Damit ist S' rekursiv, weil $S_{\mathcal{L}''}$ rekursiv und der Test $ex_j \subseteq A \subseteq L_j$ in indizierten Familien natürlich effektiv ist. Bleibt zu zeigen, daß S' im TG-FIN-K-Sinne $\mathcal{L}''$ erlernt. Es sind zwei Fälle möglich.

1. Es gibt ein i mit $ex_i \subseteq A \subseteq L_i$. Sei $S_{\mathcal{L}''}(A) = j$. Weil $\mathcal{L}''$ injektiv ist und von $S_{\mathcal{L}''}$ erlernt wird, folgt $i = j$ und damit muß der Test „$ex''_j \subseteq A \subseteq L''_j$“ erfolgreich verlaufen. An dieser Stelle ist die Injektivität von $\mathcal{L}''$ wirklich notwendig. Also $S'(A) = j$ und S' lernt richtig.

2. Für alle i gilt $ex_i \not\subseteq A$ oder $A \not\subseteq L_i$. Sei $S_{\mathcal{L}''}(A) = j$. Dieses j gibt es, weil $S_{\mathcal{L}''}$ rekursiv ist. Aber dann folgt sofort nach Definition von S', daß die Ausgabe „Es gibt keine vernünftige Hypothese“ ist. Auch hier ist die Ausgabe von S' richtig.

Also insgesamt $\mathcal{L}' \in$ TG-FIN-K$_{\mathcal{L}''}$ mit S' und $(ex''_i)_{i\geq 0}$. D.h. die Beispiele können sogar beibehalten werden.

Nun zu TG-FIN-WK $\subset$ TG-FIN. $\subseteq$ folgt sofort aus den Definitionen; zu zeigen bleibt die Ungleichheit. Die folgende indizierte Sprachklasse $\mathcal{L}'$ liefert diese. Sei $(k_i)_{i\geq 0}$ eine wiederholungsfreie Aufzählung von K. Definiere $L_i = \{k_i\}$ für alle i und setze $\mathcal{L} = (L_i)_{i\geq 0}$. Es gibt eine indizierte Familie $\mathcal{L} = \mathcal{L}'$ mit $\mathcal{L}' \in$ TG-FIN$_{\mathcal{L}}$. Setze dafür

$$L_{\langle i,j\rangle} = \begin{cases} \{i\} & \text{falls } \varphi_i(i) \text{ in } j \text{ Schritten definiert,} \\ \emptyset & \text{sonst.} \end{cases}$$

$\mathcal{L}' \in$ TG-FIN$_{\mathcal{L}}$ folgt sofort. Bleibt $\mathcal{L}' \notin$ TG-FIN-K zu zeigen: Wenn $\mathcal{L}' \in$ TG-FIN-K mit $S \in \mathcal{R}$ gälte, würde $\chi(i) =$ „$S(\{i\}) \in I\!N$?“ das Halteproblem für φ entscheiden.

□

3.5 Satz. TG-FIN-K $\cup\, \mathcal{IF} \subset$ TG-FIN-WK $\cup\, \mathcal{IF} \subset$ TG-FIN $\cup\, \mathcal{IF}$

Beweis: Hier genügt es, zwei Beispiele für die behaupteten Separationen anzugeben; die $\subseteq$-Beziehungen sind wieder offensichtlich.

Für die erste Inklusion sei $L_i = \{\langle i,0\rangle\} \cup \{\langle i,i\rangle \mid \varphi_i(i)\downarrow\}$, $ex_i = \{\langle i,0\rangle\}$ für alle i und $\mathcal{L} = (L_i)_{i\geq 0}$. Diese Sprachfamilie ist uniform r.a., aber nicht aus $\mathcal{IF}$; siehe Beispiel 2.16. Die folgende Strategie zeigt $\mathcal{L} \in$ TG-FIN-WK.

$S(A) =$ „Wenn es genau ein i mit $\langle i,0\rangle \in A$ gibt, gib i aus. Stop.
Ansonsten gib 0 aus.“

S ist rekursiv und korrekt. Wäre $\mathcal{L} \in$ TG-FIN-K, dann $\mathcal{L} \in$ TG-FIN-K$_{\mathcal{L}}$, denn $\mathcal{L}$ kann auf jedes $\mathcal{H} \in \mathcal{RA}$ mit $\mathcal{H} = \mathcal{L}$ reduziert werden. Das gilt, weil jede Sprache $L_i \in \mathcal{L} = \mathcal{H}$ wegen $\langle i,0\rangle \in L_i$ eindeutig zugeordnet werden kann. Wenn also $\mathcal{L} \in$ TG-FIN-K$_{\mathcal{H}}$, dann können die Beispiele mittels der Reduktionsfunktion auch für den Hypothesenraum $\mathcal{L}$ verwendet werden und es folgt $\mathcal{L} \in$ TG-FIN-K$_{\mathcal{L}}$. Dann gibt also gute Beispiele $(ex_i)_{i\geq 0}$ für den Hypothesenraum $\mathcal{L}$ und eine Funktion d wie in 3.3

beschrieben. Aber $\chi(i) = d(ex_i \cup \{\langle i,i\rangle\})$ würde das Halteproblem für φ entscheiden. Also folgt insgesamt $\mathcal{L} \notin$ TG-FIN-K $\cup\ \mathcal{IF}$.

Sei φ eine Numerierung von $\mathcal{P}$. Die zweistellige Kodierung von $I\!N^2$ kann leicht zu einer Kodierung von $I\!N^3$ erweitert werden. Definiere $\mathcal{L}_{\langle\rangle}$ durch

$$L_{\langle x,y,z\rangle} = \begin{cases} \{2\langle x,y,z\rangle, 2\langle x,y,z\rangle+1\} & \text{falls } \varphi_x(y)\downarrow \text{ und } \varphi_y(z)\downarrow \\ \{2\langle x,y,z\rangle\} & \text{falls } \varphi_x(y)\downarrow \text{ und } \varphi_y(z)\uparrow \\ \{2\langle x,y,z\rangle+1\} & \text{falls } \varphi_x(y)\uparrow \text{ und } \varphi_y(z)\downarrow \\ \emptyset & \text{falls } \varphi_x(y)\uparrow \text{ und } \varphi_y(z)\uparrow \end{cases}$$

Offensichtlich ist $\mathcal{L}_{\langle\rangle}$ uniform r.a. und jede Zahl kann in höchstens einer Sprache vorkommen (+).

Behauptung 1: Keine indizierte Familie $\mathcal{L}$ enthält die gleichen Sprachen.

Beweis: Angenommen, so ein $\mathcal{L}$ existiert und $\chi_{\mathcal{L}}$ entscheidet das Wortproblem für $\mathcal{L}$. Sei weiter x derart, daß φ_x rekursiv ist. Definiere

$\chi_K(y) =$ „Suche ein i mit $2\langle x,y,y\rangle \in L_i$.
Falls $2\langle x,y,y\rangle + 1 \in L_i$ gib 1 aus, sonst gib 0 aus. Stop.“

Da φ_x rekursiv ist, gilt $\varphi_x(y)\downarrow$, also $2\langle x,y,z\rangle \in L_{\langle x,y,z\rangle}$. Nach Annahme gilt $L_{\langle x,y,y\rangle} \in \mathcal{L}$ und es gibt ein i mit $2\langle x,y,y\rangle \in L_i$. D.h. der erste Schritt von χ_K terminiert immer. Wegen $\chi_{\mathcal{L}} \in \mathcal{R}$ ist auch der Test „$2\langle x,y,y\rangle+1 \in L_i$?“ effektiv und demnach χ_K eine rekursive Funktion. Es bleibt noch $\chi_K(y) = 1 \iff \varphi_y(y)\downarrow$ zu zeigen.
Sei $\varphi_y(y)\downarrow$. Dann ist $L_{\langle x,y,y\rangle} = \{2\langle x,y,y\rangle, 2\langle x,y,y\rangle + 1\}$. Also findet χ_K im ersten Teil ein i. Wegen (+) ist diese Sprache eindeutig, d.h. es gilt $L_i = L_{\langle x,y,y\rangle}$. Dann folgt $2\langle x,y,y\rangle + 1 \in L_i$ und damit $\chi_K(y) = 1$.
Gilt andererseits $\chi_K(y) = 1$, dann gibt es ein i mit $L_i = \{2\langle x,y,y\rangle, 2\langle x,y,y\rangle+1\}$. Nach Annahme folgt daraus $L_i = L_{\langle x,y,y\rangle}$. Die Definition von $\mathcal{L}_{\langle\rangle}$ liefert dann $\varphi_x(y)\downarrow$ und $\varphi_y(y)\downarrow$.
Also enstscheidet χ_K das Halteproblem für φ. $\lightning$ $\diamond$

Behauptung 2: $\mathcal{L}_{\langle\rangle} \in$ TG-FIN

Beweis: Mit normalen Methoden läßt sich eine injektive Numerierung $\mathcal{L}$ von $\mathcal{L}_{\langle\rangle}$ konstruieren, die $L_0 = \emptyset$ erfüllt. Daraus folgt insbesondere, daß jede andere Sprache aus $\mathcal{L}$ nicht leer ist. Damit ist $\mathcal{L} \in$ TG-FIN$_{\mathcal{L}}$ leicht zu zeigen. Definiere $ex_0 = \emptyset$ und für alle $i \geq 1$

$ex_i =$ „Suche in L_i nach einem x und gib $\{x\}$ zurück.“

Die Familie $(ex_i)_{i\geq 0}$ ist r.g. und die Strategie ist ebenfalls offensichtlich: Falls S als Eingabe die leere Menge bekommt, dann gibt sie 0 aus, ansonsten sucht sie eine konsistente Hypothese. Wegen (+) ist diese auch richtig. $\diamond$

Bleibt zu zeigen, daß $\mathcal{L}_{\langle\rangle}$ nicht aus TG-FIN-WK ist. Angenommen, $\mathcal{L}_{\langle\rangle} \in$ TG-FIN-WK$_{\mathcal{L}}$ mit $S \in \mathcal{R}$ und guten Beispielen $(ex_i)_{i\geq 0}$. Weil dann insbesondere $\mathcal{L}_{\langle\rangle} \in$ TG-FIN$_{\mathcal{L}}$, folgt mit 2.11, daß „$L_i = \emptyset$?“ für $\mathcal{L}$ entscheidbar ist. Definiere nun χ für alle i wie folgt.

$\chi(i) =$ „Setze $A = \{2\langle i,i,i\rangle, 2\langle i,i,i\rangle + 1\}$.
Berechne $j = S(A)$.
Falls $L_j = \emptyset$, dann gib 0 aus. Stop.
Berechne ein $x \in L_j$.
Falls $x \in A$, dann gib 1 aus. Stop.
Gib 0 aus.“

Wegen $S \in \mathcal{R}$ ist χ berechenbar und total. Es sind zwei Fälle möglich.

$\chi(i) = 1$: Dann $S(A) = j$ und es gibt ein $x \in A \cap L_j$. Beachte, daß in $\mathcal{L}_{\langle\rangle}$ zwei nichtleere Sprachen disjunkt sind ($\otimes$). Aber dann folgt sofort $A = L_j$, weil keine Sprache in $\mathcal{L}_{\langle\rangle}$ mehr als zwei Worte hat. Also $\varphi_i(i)\downarrow$ nach Definition von $\mathcal{L}_{\langle\rangle}$.

$\chi(i) = 0$: Dann $L_j = \emptyset$ oder – wegen $\otimes$ – $A \cap L_j = \emptyset$.
Dann kann es kein k mit $ex_k \subseteq A \subseteq L_k$ geben, sonst wäre S falsch. Aber dann folgt $\varphi_i(i)\uparrow$ sofort aus der Definition von $\mathcal{L}_{\langle\rangle}$.

Also entscheidet χ das Halteproblem für φ. $\lightning$
Damit insgesamt $\mathcal{L}_{\langle\rangle} \in$ TG-FIN $\setminus$ TG-FIN-WK. $\square$

Kapitel 4

Lernen mit Informantbeispielen

Zunächst eine einfache Charakterisierung von IG-LIM, die sehr der von TG-LIM ähnelt und wieder den Sätzen von Lange und Zeugmann im Aufbau folgt.

4.1 Satz. Sei $\mathcal{L}' \in \mathcal{RA}$, $\mathcal{L} = (L_i)_{i\geq 0}$ eine Aufzählung von $\mathcal{L}'$. Es gilt genau dann $\mathcal{L}' \in$ IG-LIM bezüglich $\mathcal{L}$, wenn r.g. Mengenfamilien $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ existieren, die folgende Bedingungen erfüllen.

(a) $\forall i : ex_i^+ \subseteq L_i \;\wedge\; ex_i^- \subseteq \overline{L_i}$.

(b) $\forall i,j : (ex_i^+ \subseteq L_j \;\wedge\; ex_i^- \subseteq \overline{L_j} \;\wedge\; ex_j^+ \subseteq L_i \;\wedge\; ex_j^- \subseteq \overline{L_i}) \;\Rightarrow\; L_i = L_j$.

Beweis: „$\Leftarrow$“ Seien $\mathcal{L}$, $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ entsprechend gegeben. Definiere die Strategie S wie folgt.

$S(A,B,n) =$ „Suche das minimale $z \leq n$, so daß $ex_z^+ \subseteq A \subseteq L_z$ und $ex_z^- \subseteq B$ in n Schritten beweisbar ist (mit „dove-tailing“) und weiterhin $B \cap L_z = \emptyset$ in n Schritten nicht widerlegt werden kann.
Falls z existiert, gib es aus; ansonsten gib 0 aus.“

S ist offensichtlich berechenbar. Sei i beliebig und gelte $ex_i^+ \subseteq A \subseteq L_i$, und $ex_i^- \subseteq B \subseteq \overline{L_i}$. Dann existiert ein minimales k mit $ex_k^+ \subseteq A \subseteq L_k$, $ex_k^- \subseteq B$ und $B \cap L_k = \emptyset$.

Beh.: $\lim_{n\to\infty} S(A,B,n) = k$.

Bew.: Sei $j < k$ beliebig. Nach Definition von k muß einer der folgenden Fälle eintreten.

1. $ex_j^+ \not\subseteq A$. Dann kann $ex_j^+ \subseteq A$ natürlich nie bewiesen werden und S wird nie die Hypothese j ausgeben.
2. $A \not\subseteq L_j$. Analog zu 1.
3. $ex_j^- \not\subseteq B$. Analog zu 1.
4. $B \cap L_j \neq \emptyset$. Dies werde in Schritt n_0 beweisbar. Dieses n_0 existiert, da B eine gegebene endliche Menge und L_j r.a. ist. Danach wird S allerdings die Hypothese j nie wieder ausgeben.

Also kann S gegen kein solches j konvergieren. Für L_k kann offensichtlich keiner der Fälle 1. bis 3. eintreten und wegen $B \subseteq \overline{L_k}$ kann auch Fall 4. nie auftreten. Also verläßt $S(A,B,n)$ die Hypothese k nicht mehr, wenn sie einmal ausgegeben wurde. Da nach einem *minimalen* z gesucht wird, muß S zu k kommen und konvergiert dann.

Beh.: $L_k = L_i$.

Bew.: Es gilt $ex_i^+ \subseteq A \subseteq L_i$, $ex_i^- \subseteq B \subseteq \overline{L_i}$ und $ex_k^+ \subseteq A \subseteq L_k$. Da $B \cap L_k = \emptyset$ aus der Konvergenz von S gegen k folgt, gilt $B \subseteq \overline{L_k}$ und damit insgesamt $ex_i^+ \subseteq L_k$, $ex_i^- \subseteq \overline{L_k}$, $ex_k^+ \subseteq L_i$, $ex_k^- \subseteq \overline{L_i}$. Die zweite Bedingung impliziert dann $L_i = L_k$.

Da i beliebig war, erlernt S ganz $\mathcal{L}$.

„$\Rightarrow$" Sei also $\mathcal{L}' \in$ IG-LIM bezüglich $\mathcal{L}$. Dann existieren r.g. Familien $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ wie in Definition 1.5 gefordert. Damit erfüllen sie (a).

Gelte also $ex_i^+ \subseteq L_j$, $ex_i^- \subseteq \overline{L_j}$, $ex_j^+ \subseteq L_i$ und $ex_j^- \subseteq \overline{L_i}$ für beliebige $i,j \in I\!N$. Dann folgt

$$\begin{aligned} L_i &= L_{\lim_{n\to\infty} S(ex_i^+, ex_i^-, n)} \\ &= L_{\lim_{n\to\infty} S(ex_i^+ \cup ex_j^+, ex_i^- \cup ex_j^-, n)} \\ &= L_{\lim_{n\to\infty} S(ex_j^+, ex_j^-, n)} \\ &= L_j. \end{aligned}$$

□

4.2 Korollar. Diese Charakterisierung gilt damit auch für indizierte Sprachfamilien.

4.3 Definition. Definiere $\mathcal{L}_{\overline{K}} \in \mathcal{RA}$ für eine beliebige Numerierung φ von $\mathcal{P}^1$ wie folgt.

$\mathcal{L}_{\overline{K}}$	0	1	2	3	4
L_0	„$\varphi_0(0)\downarrow$?"	1	1	1	...
L_1	1	„$\varphi_1(1)\downarrow$?"	1	1	...
L_2	1	1	„$\varphi_2(2)\downarrow$?"	1	...
L_3	1	1	1	„$\varphi_3(3)\downarrow$?"	...
	...				

D.h. für $\varphi_i(i)\downarrow$ ist $L_i = 1^\infty$, ansonsten hat L_i an der Stelle i ein „Loch". •

4.4 Lemma. $\mathcal{L}_{\overline{K}} \notin$ IG-LIM

Beweis: Sei $\mathcal{L} = (L_i)$ eine beliebige, uniform rekursive Aufzählung von $\mathcal{L}_{\overline{K}}$. Für alle i sei m_i die einzige Zahl, die eventuell in L_i fehlen kann. Weil $\mathcal{L}$ eine beliebige Aufzählung von $\mathcal{L}_{\overline{K}}$ ist, muß nicht mehr $i = m_i$ gelten.

Es genügt zu zeigen, daß es keine r.g. Familien $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ gibt, die (a) und (b) aus Satz 4.1 erfüllen.

Seien $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ beliebige r.g. Familien, die $ex_i^+ \subseteq L_i$ und $ex_i^- \subseteq \overline{L_i}$ für alle i erfüllen. Definiere nun

- $K_+ = \{i \mid \varphi_{m_i}(m_i)\downarrow\}$,
- $K_- = \{i \mid \varphi_{m_i}(m_i)\uparrow\}$,
- $K_m = \{m_i \mid i \in K_-\}$.

Dann gilt für alle $i \in K_+$: $ex_i^- = \emptyset$, da $L_i = I\!N$ und für alle $i \in K_- : \overline{L_i} = \{m_i\}$.

Beh.: $ex_i^- = \{m_i\}$ kann nicht für alle $i \in K_-$ erfüllt sein.
Bew.: Da $(ex_i^-)_{i\geq 0}$ r.g., ist $\bigcup_i ex_i^-$ r.a. Wäre $ex_i^- = \{m_i\}$ erfüllt, folgte $\bigcup_i ex_i^- = K_m$ und $\overline{K} = K_m$ wäre r.a. $\circ$

Definiere $M = \{m_i \mid ex_i^- = \emptyset \wedge i \in K_-\}$.

Beh.: M ist unendlich.

Beweis: Angenommen, M endlich; also etwa $M = \{m_1, \ldots m_p\}$. Da für alle $m_i \in K_m \setminus M$ gilt $ex_i^- \neq \emptyset$, folgt $ex_i^- = \{m_i\}$. Da $M \subset K_m$, folgt $K_m = (K_m \setminus M) \cup M$. M ist r.a., weil M endlich. Weiter gilt nun $K_m \setminus M = \bigcup_i ex_i^-$, denn

„$\subseteq$“ $x \in K_m \setminus M$ impliziert $\exists i : m_i = x \wedge ex_i^- = \{m_i\}$.

„$\supseteq$“ Sei $ex_i^- = \{m_i\}$. Daraus folgt $\varphi_{m_i}(m_i)\uparrow$ und weiter $i \in K_-$, d.h. $m_i \in K_m$. Weiter gilt nach Definition von M offensichtlich $m_i \notin M$, da $ex_i^- \neq \emptyset$. Insgesamt also $m_i \in K_m \setminus M$.

Da $\bigcup_i ex_i^-$ r.a., folgt $K_m \setminus M$ r.a. Also $(K_m \setminus M) \cup M$ r.a. und damit K_m r.a. Wegen $\overline{K} = K_m$ folgt $\overline{K}$ r.a. $\lightning$

Die Behauptung folgt. $\circ$

Betrachte ein beliebiges $j \in K_+$. Dann $ex_j^+ \subseteq I\!N$ und $ex_j^- = \emptyset$. Sei $r = max\ ex_j^+$. Da M unendlich existiert ein $\hat{m} \in M$ mit $\hat{m} > r$. Weil $\hat{m} \in M$ existiert ein $k \in K_-$ mit $m_k = \hat{m}$. Nun gelten

- $ex_j^+ \subseteq L_k$, da $L_k = 1 \ldots 1^{m_k - 1} \underbrace{\text{„}\varphi_{m_k}(m_k)\downarrow?\text{“}}_{\text{hier} = 0} 1^\infty$ und $r < m_k = \hat{m}$.
- $ex_j^- \subseteq \overline{L_k}$, da $ex_j^- = \emptyset$.
- $ex_k^+ \subseteq L_j$, da $L_j = I\!N$.
- $ex_k^- \subseteq \overline{L_j} = \emptyset$, da $m_k \in M$ und damit $ex_k^- = \emptyset$.

Nun gilt aber $L_j \neq L_k$, d.h. $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ erfüllen (b) aus Satz 4.1 nicht. Da $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ zwar r.g. gewählt wurden und (a) erfüllen, ansonsten aber beliebig sind, folgt die Behauptung aus Satz 4.1. $\square$

4.5 Satz. IG-LIM $\subset \mathcal{RA}$

Hier sind also nicht mehr alle Sprachfamilien lernbar, im Gegensatz zu IG-FIN = IG-LIM = $\mathcal{IF}$ bei indizierten Sprachfamilien. Das ist auch nicht weiter erstaunlich.

Der eigentliche Grund für Lemma 4.4 ist, daß jede Menge von Textbeispielen auf unendlich viele Sprachen paßt, so daß die Unterscheidung mittels der negativen Beispiele erfolgen müßte. Diese sind aber nun nicht berechenbar, und der Lehrer kann sie demnach nicht finden.

4.6 Korollar. IG-FIN und IG-LIM sind nicht abgeschlossen bzgl. Teilmengenbildung.

Beweis: Offensichtlich ist $\mathcal{L}_{\overline{K}}$ eine echte Teilmenge von $\mathcal{L}_1$. Weiter gilt $\mathcal{L}_1 \in$ IG-FIN und $\mathcal{L}_{\overline{K}} \notin$ IG-LIM. Damit folgt sofort die Behauptung, denn IG-FIN $\subseteq$ IG-LIM gilt nach Definition. $\square$

Nun zu IG-FIN. Leider ist mir bis jetzt keine Charakterisierung gelungen. Eine finite Strategie muß in *endlicher* Zeit zu Informationen über $\overline{L}$ kommen um die negativen Beispiele nutzen zu können, aber das ist natürlich nicht leicht. Der „Ausweg“ über den Test $ex_i^- \cap L_j = \emptyset$ ist hier nicht möglich, weil er normalerweise erst im Limes ein richtiges Ergebnis liefert.

4.7 Satz. Sei $\mathcal{L}' \in \mathcal{RA}$ derart, daß es eine *injektive* Aufzählung $\mathcal{L}$ für $\mathcal{L}'$ gibt. Dann gilt: $\mathcal{L}' \in$ IG-FIN$_{\mathcal{L}}$ $\iff$ Es gibt r.g. Familien $(ex_i^+)_{i\geq 0}$, $(ex_i^-)_{i\geq 0}$ die folgende Bedingungen erfüllen:

(1) $\forall i : ex_i^+ \subseteq L_i$ und $ex_i^- \subseteq \overline{L_i}$.

(2) $\forall i, j : ex_i^+ \subseteq L_j \ \wedge\ ex_i^- \subseteq \overline{L_j} \ \wedge\ ex_j^+ \subseteq L_i \ \wedge\ ex_j^- \subseteq \overline{L_i}$ dann $L_i = L_j$.

(3) $\exists d \in \mathcal{P}^2\, \forall k, l : (\exists i : ex_i^+ \subseteq E_k \subseteq L_i \ \wedge\ ex_i^- \subseteq E_l \subseteq \overline{L_i}) \Longrightarrow$

  i) $d(k, l)\downarrow$,

  ii) $\forall i : i \in E_{d(k,l)} \Longrightarrow ex_i^- \subseteq E_l \subseteq \overline{L_i}$,

  iii) $\exists i : i \in E_{d(k,l)} \ \wedge\ ex_i^+ \subseteq E_k \subseteq L_i \ \wedge\ ex_i^- \subseteq E_l \subseteq \overline{L_i}$.

Bedingungen (1) und (2) sind notwendig, wie schon in Satz 4.1. Bedingung (3) sieht viel komplizierter aus, als sie eigentlich ist. Anschaulich besagt sie, daß es ein berechenbares Verfahren gibt, erfolgversprechende „Hypothesenkanditaten“ zu finden; nämlich gerade die Menge $E_{d(k,l)}$: Wenn es überhaupt eine richtige Menge guter Beispiele gibt, dann nennt d eine Menge von Kandidaten, deren „negativer Beispielteil “ auf jeden Fall stimmt - ii) - und die eine komplett richtige Menge guter Beispiele enthält - iii). D.h. die Arbeit des Lehrers ist nicht prinzipiell unmöglich.

Die entscheidende Stelle ist ii), und dort der Test „$E_l \subseteq \overline{L_i}$“. Die Funktion d soll später von der Strategie verwendet werden, um Kandidaten im Hypothesenraum zu finden. Die Menge E_l ist dabei die um die Beispiele vom Gegenspieler angereicherte Menge guter Beispiele. Weil der Gegenspieler in der Regel nicht berechenbar ist, muß also d eventuell nicht berechenbare Information auf berechenbare Weise verifizieren.

Beweis: „$\Rightarrow$“ Sei also $\mathcal{L}' \in$ IG-FIN bezüglich eines injektiven $\mathcal{L}$. Dann existieren S, (ex_i^+), (ex_i^-) wie in der Definition gefordert.

Damit gilt (1) sofort. Auch (2) folgt auf die übliche Art, deshalb hier nur die „Kurzfassung“: Seien die Voraussetzungen von (2) erfüllt.

$$L_i = L_{S(ex_i^+, ex_i^-)} = L_{S(ex_i^+ \cup ex_j^+, ex_i^- \cup ex_j^-)} = L_{S(ex_j^+, ex_j^-)} = L_j$$

Zu (3). Definiere d wie folgt.

$$d(k,l) = \begin{cases} \text{Ein } m, \text{ so daß } E_m = \{S(E_k, E_l)\} & \text{falls } S(E_k, E_l) \downarrow, \\ \uparrow & \text{sonst.} \end{cases}$$

Damit ist d berechenbar.

Angenommen, es existiert ein i mit $ex_i^+ \subseteq E_k \subseteq L_i$ und $ex_i^- \subseteq E_l \subseteq \overline{L_i}$. Dann ist $S(E_k, E_l)$ definert, etwa $S(E_k, E_l) = n$. Damit ist auch $d(E_k, E_l)$ definiert und es gilt $E_{d(k,l)} = \{n\}$. Da nach Annahme $ex_i^+ \subseteq E_k \subseteq L_i$ und $ex_i^- \subseteq E_l \subseteq \overline{L_i}$ gilt, folgt $i = n$, weil $\mathcal{L}$ injektiv ist. Aber damit folgen ii) und iii) sofort.

„$\Leftarrow$“ Seien (ex_i^+), (ex_i^-) und d wie gefordert gegeben. Definiere S wie folgt.

$S(A, B) =$ „Suche k, l, so daß $E_k = A$ und $E_l = B$.
Berechne $m = d(k, l)$.
Berechne $\{n_1, \ldots, n_p\} = E_m$.
Berechne $E' = \{n_i \in E_m \mid ex_{n_i}^+ \subseteq A \text{ und } ex_{n_i}^- \subseteq B\}$. Sei etwa $E' = \{i_1, \ldots, i_q\}$.
Gib ein beliebiges i_v mit $v \leq q$ und $A \subseteq L_{i_v}$ aus, falls ein solches existiert.“

S ist damit natürlich berechenbar, weil d berechenbar ist. Sei nun $i \in I\!N$ beliebig und gelte $ex_i^+ \subseteq A \subseteq L_i$ und $ex_i^- \subseteq B \subseteq \overline{L_i}$. Gelte weiter $E_k = A$ und $E_l = B$. Dann ist nach Voraussetzung i) $d(k,l) \downarrow$, etwa $E_{d(k,l)} = \{n_1, \ldots, n_p\}$. Nach iii) ist auch E' (wie in der Definition von S) nicht leer, etwa $E' = \{i_1, \ldots, i_q\}$. Für alle n_v gilt wegen ii) $ex_{n_v}^- \subseteq E_l = B \subseteq \overline{L_{n_v}}$. Also gilt dies insbesondere für alle $i_u \in E'$. Nach iii) existiert ein $i_u \in E'$ mit $A \subseteq L_{i_u}$: i_u existiert in $E_{d(k,l)}$ mit $ex_{i_u}^+ \subseteq A \subseteq L_{i_u}$ und $ex_{i_u}^- \subseteq B \subseteq \overline{L_{i_u}}$. Dann aber auch $i_u \in E'$ nach Definition von E'. Dann ist aber $A \subseteq L_{i_u}$ beweisbar. Angenommen, S gibt i_w aus. Dann folgt sofort $ex_{i_w}^+ \subseteq A \subseteq L_{i_w}$ und $ex_{i_w}^- \subseteq B \subseteq \overline{L_{i_w}}$. Daraus folgt $L_{i_w} = L_i$ mit (2). Die Behauptung folgt, weil i beliebig war. $\square$

Auf den ersten Blick erscheint diese Charakterisierung etwas willkürlich, aber es gibt eine Reihe von Arbeiten über die Existenz von injektiven Numerierungen für uniform r.a. Sprachfamilien, z.B. [Friedberg], [Lachlan], [Kummer] etc. Da die Beweise in der Regel konstruktiv sind, können die so erhaltenen Numerierungen auf die in Satz 4.6 angegebenen Eigenschaften überprüft werden. Wie praktikabel das ist, sei dahingestellt.

Benötigt wird die Bedingung „$\mathcal{L}$ injektiv“ nur in der „$\Rightarrow$“-Richtung, und zwar, weil die Ausgabe n von S nur $L_n = L_i$ erfüllen muß, aber damit nichts über ein Enthaltensein von ex_n^+ in E_l, beziehungsweise von ex_n^- in E_k, gesagt ist. Daraus folgt nun

4.8 Korollar. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L}$ eine Aufzählung für $\mathcal{L}'$. Es gilt $\mathcal{L}' \in$ IG-FIN bezüglich $\mathcal{L}$, wenn es r.g. Familien $(ex_i^+)_{i \geq 0}$, $(ex_i^-)_{i \geq 0}$ gibt und die Bedingungen (1), (2) und (3) aus Satz 4.7 erfüllt sind.

Beweis: Die „$\Leftarrow$“-Richtung aus dem Beweis von Satz 4.7 kann ohne Änderungen übernommen werden. $\square$

Angenommen, der Lehrer erfüllt folgende Bedingung: Wenn $L_i = L_j$, so auch $ex_i^+ = ex_j^+$ und $ex_i^- = ex_j^-$. Ist diese Forderung realistisch? Die Antwort darauf ist einmal mehr ein klares „jein“. Einerseits ist es effizient, für gleiche Sprachen die gleichen Beispiele zu verwenden. Außerdem verwirrt es die Schüler wahrscheinlich weniger. Andererseits ist vielleicht in der Codierung i bzw. j auf verschiedene Aspekte der Sprache Wert gelegt worden, so daß den Indizes eine weitere als nur kennzeichnende Bedeutung zukommt.

4.9 Definition. Sei $\mathcal{L} \in \mathcal{RA}$ und seien $(ex_i^+)_{i\geq 0}$, $(ex_i^-)_{i\geq 0}$ r.g. Familien mit $ex_i^+ \subseteq L_i$ und $ex_i^- \subseteq \overline{L_i}$ für alle i. Diese Familien werden *einheitlich* genannt, falls $L_i = L_j$ auch $ex_i^+ = ex_j^+$ und $ex_i^- = ex_j^-$ für alle i, j impliziert. $\bullet$

4.10 Korollar. Sei $\mathcal{L}' \in \mathcal{RA}$ und $\mathcal{L}$ eine Aufzählung für $\mathcal{L}'$. Dann gilt: $\mathcal{L}' \in$ IG-FIN$_\mathcal{L}$ mit einheitlichen guten Beispielen $\iff$ Es gibt *einheitliche* r.g. Familien $(ex_i^+)_{i\geq 0}$, $(ex_i^-)_{i\geq 0}$ für die die Bedingungen (1), (2) und (3) aus Satz 4.7 erfüllt sind.

Beweis: “$\Rightarrow$“ Der Beweis läuft genau wie der von Satz 4.7, nur wird hier nicht $i = n$ aus der Injektivität von $\mathcal{L}$ geschlossen, sondern $ex_i^+ = ex_n^+$ und $ex_i^- = ex_n^-$ aus der Einheitlichkeit der guten Beispiele. Aber damit folgen ii) und iii) wieder sofort.

„$\Leftarrow$“ Folgt sofort aus Korollar 4.8. $\square$

Damit ist 4.10 leider keine Charakterisierung von IG-FIN, weil IG-FIN natürlich allgemeiner als „IG-FIN mit einheitlichen guten Beispielen“ definiert ist.

Bis jetzt steht allerdings die Frage im Raum, ob die negativen Beispiele überhaupt einen Vorteil bringen. Die Antwort für TG-FIN und IG-FIN für r.a. Sprachfamilien folgt allerdings aus der Antwort für indizierte Sprachfamilien, siehe [LNW].

4.11 Lemma. IG-FIN $\supset$ TG-FIN

Beweis: $\mathcal{L}_1$ liegt in IG-FIN $\setminus$ TG-FIN; TG-FIN $\subseteq$ IG-FIN ist offensichtlich. $\square$

Allerdings ist $\mathcal{L}_1 \in \mathcal{IF}$ und damit stellt sich die Frage, ob es auch r.a. Familien aus IG-FIN $\setminus$ TG-FIN gibt, die nicht aus $\mathcal{IF}$ sind. Das folgende Lemma beantwortet diese einfache Frage positiv.

4.12 Lemma. IG-FIN $\supset$ (TG-FIN $\cup\, \mathcal{IF}$)

Beweis: IG-FIN $\supseteq$ (TG-FIN $\cup\, \mathcal{IF}$) ist offensichtlich. Bleibt „$\neq$“ zu zeigen. Dazu wird einfach eine Sprachklasse aus $\mathcal{IF} \setminus$ TG-FIN so geändert, daß sie aus $\mathcal{RA} \setminus \mathcal{IF}$ ist, ohne ihre anderen Eigenschaften zu verlieren.

$\mathcal{L}$	0	1	2	3	4	5	6
L_0	1	1	1	1	...		
L_1	1	0	0	„$\varphi_1(1)\downarrow$?“	0	0	...
L_2	1	1	0	0	„$\varphi_2(2)\downarrow$?“	0	...
L_3	1	1	1	0	0	„$\varphi_3(3)\downarrow$?“	...
	...						

Damit enthält $\mathcal{L}$ einen „Häufungspunkt" und ist nicht aus TG-FIN. Weiter ist $\mathcal{L}$ offensichtlich aus IG-FIN; wähle z.B. $ex_0^- = \emptyset$ und $ex_i^- = \{min\,\overline{L_i}\}$ für $i \geq 1$. $\mathcal{L} \in \mathcal{RA}$ braucht wohl nicht mehr gezeigt werden. Bleibt $\mathcal{L} \notin \mathcal{IF}$ zu zeigen. Aber das ist leicht: Wenn es eine $\mathcal{IF}$ für $\mathcal{L}$ gäbe, wäre K_φ entscheidbar. □

Bei indizierten Familien ist IG-FIN „allmächtig", d.h. jede indizierte Sprachfamilie ist anhand guter Informantbeispiele finit erlernbar.

4.13 Lemma. Für indizierte Sprachklassen ist CTG-FIN echt weniger als IG-FIN: CTG-LIM $\cap\, \mathcal{IF} \subset$ IG-FIN $\cap\, \mathcal{IF}$.

Der Beweis ist sehr einfach: Jede indizierte Sprachfamilie liegt in IG-FIN, siehe [LNW]. Weil CTG-FIN keine Sprachklassen enthält, die alle endlichen und mindestens eine unendliche Sprache enthalten, folgt CTG-LIM $\subset \mathcal{IF}$ und damit die Behauptung.

Der folgende Satz beschreibt nun einen weiteren Unterschied zwischen indizierten und uniform r.a. Sprachfamilien.

4.14 Satz. CTG-FIN # IG-LIM

Beweis: Es gelten

- $\mathcal{L}_1 \in$ IG-LIM,
- $\mathcal{L}_1 \notin$ CTG-LIM, siehe Beweis von Satz 2.12.
- $\mathcal{L}_\Phi \in$ CTG-FIN.

Es bleibt nur noch $\mathcal{L}_\Phi \notin$ IG-LIM zu zeigen.

Angenommen, es gilt $\mathcal{L}_\Phi \in$ IG-LIM bzgl. eines Hypothesenraumes $\mathcal{L}$ mit $\mathcal{L}_\Phi = \mathcal{L}$ und mit r.g. guten Beispielen $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$, die (a) und (b) aus Satz 4.1 erfüllen. Es sind zwei Fälle möglich.

Fall 1: Es gibt nur endlich viele $ex_i^- \neq \emptyset$.

Sei i maximal mit $ex_i^- \neq \emptyset$. Bilde dann eine streng monoton wachsende Folge aus den $max\; ex_j^+$, $j \geq i$. Diese Folge führt wie im Beweis von Satz 2.12 zum Widerspruch.

Fall 2: Es gibt unendlich viele $ex_i^- \neq \emptyset$.

Bilde dann eine streng monoton wachsende Folge aus den $max\; ex_i^-$. Wenn $ex_i^- \neq \emptyset$, dann gilt $max\; ex_i^- > max\; L_i$. Aus dieser Folge ergäbe sich die Entscheidbarkeit des Halteproblems wie in 2.12. □

$\mathcal{L}_\Phi$ ist anscheinend *das* Argument für die C-Typen: nicht aus IG-LIM, aber aus CTG-FIN. Dabei ist es eine Klasse, die von vorneherein nichts mit Lernen zu tun hat. Die Schwierigkeit dieser Klasse besteht darin, daß die maximalen Worte der Sprachen eine nicht rekursiv aufzählbare Menge bilden.

Andererseits hat dieses Resultat aber einen Schönheitsfehler. Sei nämlich $\mathcal{L}_\Phi = (L_i)_{i\geq 0}$. Zwar gilt $\mathcal{L}_\Phi \in$ CTG-FIN$_{\text{FINITE}}$, aber es gibt keinen rekursiven Compiler $c : \mathcal{L}_\Phi \mapsto$

FINITE mit $L_i = E_{c(i)}$ für alle i. D.h. jedes L_i kommt in FINITE vor, aber die Stelle ist unbekannt. Es ist also in gewisser Weise unmöglich, L_3 zu erlernen, weil der Lehrer nicht weiß, wo L_3 in FINITE vorkommt. Der Lehrer kann also „Lehraufträge" nur bezüglich FINITE, aber nicht mehr bezüglich dem ursprünglichen Lernziel – der Klasse $\mathcal{L}_\Phi$ – ausführen. Wenn gerade die Klasse $\mathcal{L}_\Phi$ interessiert, dann ist das Lernziel trotz des formalen Erfolges m.E. nicht erreicht worden.

Allerdings gibt es einen limesrekursiven Compiler, der beliebige $\mathcal{L}_\Phi$-Nummern in äquivalente FINITE-Nummern übersetzt, so daß der Lehrer seinen Lehrauftrag bzgl. $\mathcal{L}_\Phi$ im Hypothesenraum FINITE wenigstens im Limes ausführen kann.

Nun zu einer anderen Frage. In Satz 4.1 wird gezeigt, daß eine Strategie alle IG-LIM-Lernprobleme lösen kann. In [LNW] wird eine sehr ähnliche Idee verwendet, um alle Sprachfamilien aus $\mathcal{IF}$ zu lernen. Gibt es auch „universelle" Beispiele für die Sprachfamilien, so daß diese Beispiele so typisch für die Sprachen sind, daß sie in allen Hypothesenräumen eingesetzt werden können? Insbesondere sollten diese Beispiele dann einheitlich im Sinne von Definition 4.9 sein.

Für $\mathcal{RA}$ gibt es diese Beispiele im allgemeinen nicht, wie Satz 2.22 belegt; einheitliche gute Beispiele implizieren ja die Entscheidbarkeit der Gleichheit in allen Hypothesenräumen.

Für $\mathcal{IF}$ ist die Frage nicht ganz so einfach zu beantworten, besagt doch Theorem 9 aus [LNW], daß in jedem Hypthesenraum für indizierte Familien die Gleichheit entscheidbar ist, und dies widerspricht der Einheitlichkeit der guten Beispiele nicht, es legt sogar die Vermutung nahe, daß es solche universellen guten Beispiele geben könnte.

Definition. Sei $\mathcal{L} \in \mathcal{IF}$. Dann sei $Hyp(\mathcal{L}) = \{\mathcal{L}' \in \mathcal{IF} \mid \mathcal{L} \in \text{IG-FIN}_{\mathcal{L}'}\}$. •

Daraus folgt $Hyp(\mathcal{L}) \neq \emptyset$ für alle $\mathcal{L} \in \mathcal{IF}$.

Definition. $\mathcal{L} \in \mathcal{IF}$ heißt mit *universellen guten Beispielen finit erlernbar*, wenn ein $\mathcal{L}' \in Hyp(\mathcal{L})$ und gute Beispiele $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ existieren, so daß $\mathcal{L} \in \text{IG-FIN}_{\mathcal{L}'}$ mit den Beispielen $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$; des weiteren gibt es für alle $\mathcal{H} \in Hyp(\mathcal{L})$ r.g. Familien guter Beispiele $(hex_i^+)_{i\geq 0}$ und $(hex_i^-)_{i\geq 0}$, so daß $\mathcal{L} \in \text{IG-FIN}_{\mathcal{H}}$ mit eben diesen Beispielen und für alle i und j gilt: Wenn $H_i = L'_j$, so $hex_i^+ = ex_j^+$ und $hex_i^- = ex_j^-$. •

4.15 Satz. $\mathcal{L} \in \mathcal{IF}$ ist mit universellen guten Beispielen finit erlernbar $\iff$ Für alle $\mathcal{L}'$, $\mathcal{L}'' \in Hyp(\mathcal{L})$ gibt es ein $c \in \mathcal{R}$ mit $L'_i = L''_{c(i)}$ für alle i.

Mit anderen Worten, eine indizierte Sprachfamilie ist genau dann mit universellen Beispielen lernbar, wenn alle Hypothesenräume aufeinander reduzierbar sind.

Beweis: „$\Rightarrow$": Die Reduktion c sucht einfach nach einer Sprache mit den gleichen Beispielen.
„$\Leftarrow$": $Hyp(\mathcal{L})$ enthält nach Theorem 9 aus [LNW] jede injektive Numerierung von $\mathcal{L}$. Weil nach Voraussetzung jeder andere Hypothesenraum darauf reduzierbar ist, können die Beispiele eines injektiven Raumes auch mittels der Reduktionsfunktion für alle anderen Räume verwendet werden: Angenommen, $\mathcal{L}$ ist ein injektiver Hypothesenraum und $(ex_i^+)_{i\geq 0}$, $(ex_i^-)_{i\geq 0}$ sind die guten Beispiele dafür. Sei weiter $\mathcal{H}$ ein beliebiger anderer Hypothesenraum für $\mathcal{L}$ und $c \in \mathcal{R}$ reduziere $\mathcal{H}$ auf $\mathcal{L}$. Definiere $hex_i^+ = ex_{c(i)}^+$ und

$hex_i^- = ex_{c(i)}^-$ für alle i. Weil $(ex_i^+)_{i\geq 0}$ und $(ex_i^-)_{i\geq 0}$ – wegen Korollar 4.2 und IG-LIM = IG-FIN für indizierte Sprachfamilien, siehe [LNW] – Satz 4.1 erfüllen, tun es auch die entsprechenden hex_i-Familien. Damit kann auch im Hypothesenraum $\mathcal{H}$ mit diesen Beispielen gelernt werden. Die Behauptung folgt. □

Der nächste Satz zeigt nun, daß es eine indizierte Sprachfamilie $\mathcal{L}$ gibt, für die $Hyp(\mathcal{L})$ keine Äquivalenzklasse unter der Reduktion ist. Aus Satz 4.15 folgt dann, daß es nicht für alle indizierten Familien universelle gute Beispiele gibt.

4.16 Satz. Es gibt indizierte Familien $\mathcal{L}, \mathcal{H}$ mit

1) $\mathcal{L} = \mathcal{H}$ und beide Numerierungen sind injektiv,

2) $\mathcal{L} \in$ IG-FIN$_{\mathcal{L}}$ und $\mathcal{L} \in$ IG-FIN$_{\mathcal{H}}$,

3) es gibt keinen rekursiven Compiler $c : \mathcal{L} \mapsto \mathcal{H}$.

Beweis: Die Konstruktion der Numerierungen ist eine Diagonalisierung gegen alle rekursiven Compiler. Dabei wird der c-te Compiler auf die endliche Sprache 1^c0^∞ angewendet. Diese Sprache wird dann in $\mathcal{H}$ an eine Stelle numeriert, die c nicht herausbekommen kann. Wenn c trotzdem einen Tip abgibt, wird dafür gesorgt, daß er eine Sprache ungleich 1^c0^∞ beschreibt. Diese Sprache muß dann noch in $\mathcal{L}$ aufgenommen werden, um die Gleichheit der Familien zu sichern. Die Präfixe 1^c sichern im wesentlichen die Injektivität der beiden Familien. Nun die formale Definition. Sei φ eine Gödelnumerierung. $\mathcal{L} = (L_i)_{i\geq 0}$, $\mathcal{H} = (H_i)_{i\geq 0}$. RS ist die Menge der „richtigen Sprachen“ in $\mathcal{L}$; d.h. die Sprachen, die zur Widerlegung der Compiler konstruiert, und nicht wegen der Erhaltung der Gleichheit hinzugenommen werden. In l_i wird gespeichert, welcher Compiler mit der Sprache L_i widerlegt werden soll. Zwar kann diese Information auch aus L_i gewonnen werden, aber so bleibt die Notation übersichtlicher. Mit n wird der nächste, noch nie verwendete Index in $\mathcal{L}$ und $\mathcal{H}$ bezeichnet. Die Idee zu dieser Konstruktion stammt von R. Wiehagen. Die beiden Numerierungen werden parallel in Stufen definiert.

Initialisierung: Für alle i setze $L_i = H_i = \emptyset$ und $l_i = 0$. Setze $RS = \emptyset$ und $n = 0$.

Stufe s: Setze $L_n = 1^s0^\infty$, $l_n = s$, $H_n = 1^s0$, $RS = RS \cup \{n\}$, $n = n + 1$.
for $i = 1$ to $n - 1$, $i \in RS$ do:
Berechne $j = \varphi_{l_i}(i)$ für s Schritte, wenn H_i noch nicht abgeschlossen ist.

(1) Falls j noch undefiniert, dann $H_i = H_i0$.

(2) Falls $j = i$, dann

- setze $H_i = H_i1^\infty$, schließe H_i ab,
- setze $L_n = H_i$, $H_n = L_i$, $n = n + 1$;

(3) Falls $j \neq i$, dann setze $H_i = H_i0^\infty$ und schließe H_i ab.

next i.

Lemma: Es gelten die folgenden Aussagen.

i) $\mathcal{L}$ und $\mathcal{H}$ sind indizierte Familien und $\mathcal{L} = \mathcal{H}$.

ii) $\mathcal{L}$ und $\mathcal{H}$ sind injektiv.

iii) $\{j \mid \exists i \geq 0 : l_i = j\} = I\!N$.

Beweis: Zu i). „ $\subseteq$ “. Wenn $i \in RS$, dann $L_i = 1^{l_i}0^\infty$. Falls $\varphi_{l_i}(i) \uparrow$, dann wächst H_i in (1) zu $1^{l_i}0^\infty$. Ist $\varphi_{l_i}(i) \downarrow$ und $\varphi_{l_i}(i) \neq i$, dann wird H_i in (3) auf $1^{l_i}0^\infty$ gesetzt und abgeschlossen.

Wenn $i \notin RS$, dann wird L_i in (2) definiert. Aber dann wird L_i auf ein H_n gesetzt.

“ $\supseteq$ “. Entweder wächst H_i zu einem L_i heran in (1) oder wird in (3) auf ein L_i gesetzt. Wird H_i in (2) definert, so werden auch in $\mathcal{L}$ entsprechende Sprachen aufgenommen.

Beide Familien sind rekursiv, weil in $\mathcal{L}$ immer gleich die ganze Sprache definiert wird und in $\mathcal{H}$ die ersten s Sprachen mindestens bis zur Stelle s bekannt sind, und s über alle Grenzen wächst.

Zu ii). Es gibt in $\mathcal{L}$ und $\mathcal{H}$ für jedes s jeweils höchstens zwei Sprachen, die mit 1^s beginnen, und jede Sprache beginnt mit 1^s für ein geeignetes s. Wenn es zwei gibt, dann ist eine endlich – sie wird mit „0^∞“ abgeschlossen – und die andere unendlich – sie wird mit „1^∞“ abgeschlossen; siehe (2) in der Definition der Stufen. Also unterscheiden sich auch diese beiden Sprachen.

Zu iii). Offensichtlich wird in jeder Stufe s ein l_n auf s gesetzt. ∘

Damit ist Behauptung 1) aus dem Satz gezeigt. 2) folgt aus Theorem 7 in [LNW] denn es impliziert folgende Aussage: Sei $\mathcal{L} \in \mathcal{IF}$. Dann $\mathcal{L} \in \text{IG-FIN}_\mathcal{L} \iff$ In $\mathcal{L}$ ist die Gleichheit entscheidbar. (Der Beweis ist analog zu denen von 2.19 und 2.20.) Weil die eben konstruierten Numerierungen injektiv sind, ist in ihnen die Gleichheit natürlich entscheidbar.

Bleibt 3) zu zeigen. Angenommen, es gibt einen rekursiven Compiler $\varphi_c : \mathcal{L} \mapsto \mathcal{H}$. Nach iii) aus dem Lemma gibt es also ein i mit $l_i = c$. Betrachte φ_{l_i}: Weil φ_{l_i} rekursiv ist, gilt $\varphi_{l_i}(i) = j \in I\!N$. Es sind zwei Fälle möglich:

$j = i$: Dann $H_i = 1^{l_i}0^k1^\infty$ für ein geeignetes k, denn die Konstruktion wird in (2) entsprechend abgeschlossen. Weil $L_i = 1^{l_i}0^\infty$ folgt also $L_i \neq H_i$. ↯

$j \neq i$: Dann $L_i = H_i = 1^{l_i}0^\infty$ nach Fall (3) der Konstruktion. Weil $\mathcal{H}$ injektiv ist, folgt also $H_j \neq H_i$. ↯

Also gibt es keinen solchen Compiler und die Behauptung 3) ist bewiesen. □

Die Konstruktion läßt noch eine weitere erstaunliche Beobachtung zu. Als erstes ist zu bemerken, daß $\mathcal{L}$ und $\mathcal{H}$ jeweils selbst als Hypothesenraum gewählt werden können. Wie jetzt auch immer die Beispiele gewählt werden, stets gibt es unendlich viele i, so daß die Mengen guter Beispiele für L_i und H_i „gleich“ sind, obwohl $L_i \neq H_i$ gilt. Weil $\mathcal{L} = \mathcal{H}$

ist, gibt es also ein $L_j = H_i$ und die Beispiele, die für $\mathcal{L}$ die Sprache L_i beschreiben, beschreiben in $\mathcal{H}$ die Sprache L_j. Mit anderen Worten, die beiden Schüler lernen aus „gleichen" Beispielen etwas Verschiedenes, obwohl sie die gleiche Lernpotenz haben. Mit „gleich" ist hier nicht die syntaktische Gleichheit gemeint, sondern die Beispielmengen zweier Sprachen heißen hier gleich, wenn die „$^+$"-Beispiele einer Sprache in der jeweils anderen Sprache und die „$^-$"-Beispiele im Komplement der jeweils anderen Sprache enthalten sind. Genau das drückt (1) im folgenden Korollar aus.

4.17 Korollar. Seien $\mathcal{L}$ und $\mathcal{H}$ die indizierten Familien aus 4.16. Gelte weiter $\mathcal{L} \in$ IG-FIN$_\mathcal{L}$ mit guten Beispielen $(lex_i^+)_{i\geq 0}$, $(lex_i^-)_{i\geq 0}$ und $\mathcal{H} \in$ IG-FIN$_\mathcal{H}$ mit guten Beispielen $(hex_i^+)_{i\geq 0}$, $(hex_i^-)_{i\geq 0}$. Dann gibt es unendlich viele i, so daß gilt:

(1) $lex_i^+ \subseteq H_i$, $hex_i^+ \subseteq L_i$, $lex_i^- \subseteq \overline{H_i}$, $hex_i^- \subseteq \overline{L_i}$ und

(2) $H_i \neq L_i$.

Anschaulich besagt (1), daß die Strategie, die $\mathcal{L}$ bzgl. des Hypothesenraumes $\mathcal{L}$ lernt, bei Eingabe von hex_i^+ und hex_i^- die Sprache L_i lernen würde, die aber ungleich der Sprache H_i ist, für die diese Beispiele ursprünglich gedacht waren. Wegen $\mathcal{L} = \mathcal{H}$ lernt die $\mathcal{L}$-Strategie natürlich auch – allerdings unter anderem Namen – die Sprache H_i, aber die Beispiele dafür unterscheiden sich denmach „wesentlich" von denen, mit denen die $\mathcal{H}$-Strategie H_i lernt, denn die werden ja für L_i gebraucht. Umgekehrt würde die $\mathcal{H}$-Strategie aus lex_i^+ und lex_i^- die Sprache H_i lernen. Also würden die Schüler die Beispiele für den jeweils anderen auf verschiedene Weise interpretieren und aus ihnen verschiedene Sprachen lernen.

Beweis: Vor dem formalen Beweis, der sehr stark auf der Konstruktion aufbaut, kurz der eigentliche Grund für das Korollar. Wenn $L_i \neq H_i$, dann gibt es ein j mit $L_i = H_j$ und $H_i = L_j$. Die Menge $\{i \mid L_i = H_i\}$ ist unentscheidbar, und damit auch die Frage, ob es dieses L_j in $\mathcal{L}$ gibt. Die Sprachen L_i und H_i sind am Anfang gleich, und weil nicht bekannt ist, ob sie sich später einmal untescheiden, müssen die guten Beispiele – einmal für $\mathcal{L}$ und einmal für $\mathcal{H}$ – unendlich oft aus diesem „gleichen Teil" genommen werden. Gibt es später noch dieses L_j, d.h. wird der Unterschied bekannt, kann er in dieser zweiten Sprache – wieder für $\mathcal{L}$ und $\mathcal{H}$ – gesucht werden. Nun unterscheiden sich also L_i und H_i, aber die guten Beispiele wurden aus dem übereinstimmenden Teil gewählt und das ergibt die Behauptung.

Es werden die Bezeichnungen aus 4.16 verwendet. Als erstes sei an *iii)* aus dem Lemma in 4.16 erinnert: $\{j \mid \exists i : l_i = j\} = I\!N$. $I = \{i \mid l_i > 0 \text{ und } \varphi_{l_i}(i) = i\}$ ist nicht rekursiv. (Der Beweis dieser Aussage ist eine einfache Diagonalisierung: Angenommen, $g \in \mathcal{R}$ entscheidet I. Definiere $f(x) = x + 1$, falls $g(x) = 1$; sonst sei $f(x) = x$. Damit ist $f \in \mathcal{R}$. Sei $k > 0$ mit $\varphi_k = f$. Nach Lemma, *iii)* im Beweis von 4.16 gibt es dann ein n mit $l_n = k$.
1. Fall: $g(n) = 1 \Rightarrow f(n) = n + 1 \Rightarrow \varphi_{l_n}(n) = n + 1$ ↯ weil $l_n > 0$.
2. Fall: $g(n) \neq 1 \Rightarrow f(n) = n \Rightarrow \varphi_{l_n}(n) = n$ ↯ weil $l_n > 0$.)

Laut Konstruktion in 4.16 gilt $l_0 = 0$. Seien die Voraussetzungen aus dem Korollar erfüllt. Definiere nun für alle i:

$\chi(i) =$ „Wenn $l_i = 0$, dann gib 0 aus, Stop.
Berechne $m = max\{max\ lex_i^+, max\ hex_i^-\}$.
Berechne $\varphi_{l_i}(i)$ für m Schritte.
Falls das Ergebnis definert und gleich i ist, dann gib 1 aus, Stop.
Sonst gib 0 aus. Stop."

Damit ist χ eine rekursive Funktion, weil die Folge der l_i effektiv berechenbar ist und die guten Beispiele r.g. sind. Nun wird gezeigt, daß χ die Menge I entscheidet, wenn (1) und (2) aus der Behauptung des Korollars nicht gelten.

Also angenommen, es gibt kein i, das (1) und (2) erfüllt.

Wenn $L_i = H_i$, $i > 0$, dann gilt nach Konstruktion von $\mathcal{L}$ und $\mathcal{H}$, daß $l_i \neq 0$ ist. Weiter folgt, daß entweder $\varphi_{l_i}(i) \uparrow$ oder $\varphi_{l_i}(i) \neq i$ gilt, und damit $i \notin I$. In beiden Fällen liefert aber $\chi(i)$ das richtige Ergebnis 0, unabhängig davon, wieviele Schritte $\chi(i)$ auf $\varphi_{l_i}(i)$ rechnet.

Sei nun $L_i \neq H_i$. Wenn $L_i = 1^c0^x1^\infty$, dann $l_i = 0$. Das gilt, weil ein unendliches L_i nur in Schritt (2) der Konstruktion dazukommt, anschließend n erhöht, aber l_i nie mehr geändert wird und damit auf seinem Initialwert 0 bleibt. Also gibt χ auch in den Fällen $L_i \neq H_i$ und $L_i = 1^c0^x1^\infty$ die richtige Antwort 0.

Es bleiben damit noch die Fälle $L_i \neq H_i$, $i > 0$, und $L_i = 1^c0^\infty$. Dann wurde L_i zu Beginn einer Stufe s definiert und damit auch l_i auf s gesetzt. Weil $i > 0$ kann dies augenscheinlich nicht in Stufe 0 geschehen sein und damit gilt $l_i > 0$. Weiter ist $H_i = 1^c0^x1^\infty$, wobei $x = \Phi_{l_i}(i)$. Weil damit $L_i \subseteq H_i$ gilt, folgt unmittelbar aus den Eigenschaften der guten Beispiele $lex_i^+ \subseteq H_i$ und $hex_i^- \subseteq \overline{L_i}$. Weil nach Annahme kein i sowohl (1) und (2) erfüllt und $L_i \neq H_i$ gilt, muß $lex_i^- \not\subseteq \overline{H_i}$ oder $hex_i^+ \not\subseteq L_i$ sein (+).

Es kann nur $lex_i^- \not\subseteq \overline{H_i}$ gelten, wenn $max\ lex_i^- \geq c + x$ ist.

Es kann nur $hex_i^+ \not\subseteq L_i$ gelten, wenn $max\ hex_i^+ \geq c + x$ ist.

Damit ist $m = max\{max\ lex_i^-, max\ hex_i^+\} \geq c+x \geq \Phi_{l_i}(i)$, denn einer der beiden Fälle muß eintreten. Also berechnet $\chi(i)$ lange genug $\varphi_{l_i}(i)$ und gibt die richtige Antwort 1 aus.

Weitere Fälle sind nicht möglich und χ entscheidet I. $\lightning$

Die Annahme wurde nur an der Stelle (+) verwendet. Also gibt es wenigstens ein i mit (1), (2), $L_i = 1^c0^\infty$ und $H_i = 1^c0^x1^\infty$. Daraus folgt sofort, daß es unendlich viele solche i geben muß: Endlich viele Ausnahmen könnten in den guten Beispielen durch ein Tabelle „repariert" werden. $\square$

Kapitel 5

Diskussion und offene Fragen

Die vorgestellten Resultate sind, bis auf ein oder zwei Ausnahmen, nicht unerwartet gekommen. Einerseits ist das schade, denn wer findet oder liest nicht gerne etwas Überraschendes. Andererseits spricht es dafür, daß das Modell recht gut die ihm zugrunde liegende Idee beschreibt. Das Lernen mit Lehrer und Schüler ist uns allen gut bekannt. Wenn also das Modell nur wenige unserer Erfahrung zuwiderlaufende Resultate liefert, spricht das für die formale Umsetzung. Dann dürften sich im formalen Modell gefundene neue Gedanken so oder so ähnlich hoffentlich auch im normalen Leben wiederfinden.

In der Arbeit wurde versucht, die Separationen ohne „unnatürliche" Sprachklassen zu beweisen, sondern mit solchen, die bekannte Probleme beschreiben. Wenn eine echte Inklusion mittels einer „künstlich konstruierten" Sprachklasse bewiesen wurde, muß die Frage erlaubt sein, ob sich der größere Inferenztyp überhaupt lohnt, wenn in ihm nur Klassen liegen, die niemand je lernen wird. John Case argumentiert sehr überzeugend, daß die meisten „künstlichen" Klassen nur Vorboten „natürlicher" Klassen seien, siehe [Case]. Trotzdem sind für eine Reihe von Inferenztypen meines Wissens keine „natürlichen" Klassen bekannt, die nicht auch in einem viel einfacheren Inferenztyp liegen – obwohl es natürlich solche Klassen geben kann. Mit der gewählten Vorgehensweise sollte von vorneherein versucht werden, den Nutzen der Inferenztypen auf bekannten Problemen zu begründen. Wie gut das gelungen ist, mag der Leser entscheiden.

Die interessanteste offene Frage ist zur Zeit, ob IG-FIN $\subset$ IG-LIM gilt. Obwohl das sehr wahrscheinlich ist, scheitert ein Beweis im Moment an den Mengen A und B, die ein böswilliger Gegner der Strategie zusammen mit den guten Beispielen gibt. Wenn B aus dem Komplement der Sprache stammt, dann kann eine Strategie im Prinzip immer mit nicht uniform(!) berechenbaren Eingaben gefüttert werden. Die Limesstrategie kann die Konsistenz von B mit dem Komplement einer Sprache im Limes feststellen, aber der Beweis, daß eine finite Strategie dieses Manko in keinem Hypothesenraum ausgleichen kann, scheitert bis zu dem Tag, an dem diese Zeilen geschrieben werden.
Wenn die echte Inklusion gilt, untermauert sie die folgende These. *Wenn Klassen von Objekten anhand guter Beispiele in Hypothesenräumen gelernt werden sollen, die genau die zu lernenden Objekte enthalten, dann gilt: Entsprechender FIN-Typ = entsprechender LIM-Typ* $\iff$ *Die Konsistenz beliebiger endlicher Teilmengen der Objekte mit*

beliebigen Objekten ist uniform rekursiv aufzählbar. Wenn IG-FIN = IG-LIM gilt, ist die These widerlegt; die Konsistenz beliebiger Teilmengen des Komplementes rekursiv aufzählbarer Sprachen mit eben diesem Komplement ist natürlich nicht uniform rekursiv aufzählbar.

Auch Charakterisierungen von CTG-FIN und CTG-LIM fehlen noch. Sie sind schwieriger als etwa die von TG-FIN, weil die Mengen ex_i nicht mehr für *alle* i endlich und rekursiv generierbar sein müssen, sondern nur noch für *diejenigen* i, für die L_i eine zu erlernende Sprache ist.

Als echtes Problem hat sich der nicht berechenbare Gegenspieler erwiesen, der bei rekursiv aufzählbaren Sprachen seine ganze Kraft erstmals voll ausspielen kann. Sowohl bei den rekursiven Funktionen als auch bei den indizierten Sprachfamilien bereiten die zusätzlichen Beispiele wenig Kopfzerbrechen; sie werden einfach „mitgenommen". Die Konsistenztests der Strategien werden durch die zusätzlichen Beispiele vor keine neuen Probleme gestellt. Das ist bei uniform r.a. Sprachen anders: Hier stellt der *Gegenspieler* – und nicht der Lehrer! – die Strategien vor die Aufgabe, mit nicht berechenbaren Daten zu arbeiten.
Macht der Gegenspieler nicht das ganze Modell unbrauchbar? Schließlich werden nie „nicht berechenbare" Maschinen gebaut werden. Zwei Argumente seien hier zur Verteidigung angeführt. Einmal wäre es möglich, daß der Gegenspieler zwar berechenbar, aber auch so kompliziert ist, daß keine vernünftige Möglichkeit besteht, ihn zu simulieren. Als Beispiel könnte ein analoges neuronales Netz dienen. Zweitens ist natürlich für jedes $\mathcal{L} \in \mathcal{RA}$ jede endliche Teilmenge eines Komplementes berechenbar. Sie sind eventuell nur nicht *uniform* berechenbar. Die vom Gegenspieler hinzugefügten Beispiele sind also durchaus berechenbar, nur eben nicht für alle Sprachen mit *einem* Algorithmus.
Ein weiteres schönes Argument, warum das Arbeiten mit nicht berechenbaren Daten nicht sinnlos ist, liefert [Rogers] auf den Seiten 130 ff. Im wesentlichen argumentiert er, daß jedes Anfangsstück der Länge n einer totalen Funktion f mit Wertebereich $\{0, 1\}$ in dem vollständigen Binärbaum der Tiefe n enthalten ist, unabhängig von der Berechenbarkeit von f. Soll jetzt ein Problem bearbeitet werden, zu dessen Lösung f nötig ist, dann kann das Problem ja für alle Äste des Binärbaumes bearbeitet werden, in der Annahme, der jeweilige Ast sei das Anfangsstück von f. Wenn nun irgendwann Teile von f bekannt werden, dann ist ein Teil der Berechnungen richtig und kann beibehalten werden, andere bisher geleistete Arbeit kann zusammen mit vielen Ästen des Baumes verworfen werden.

Andererseits ist es nötig, den Gegenspieler zu verwenden und beliebig große Mengen von (Gegen-)Beispielen zuzulassen. Wird nämlich die Anzahl der (Gegen-)Beispiele auf 1 begrenzt, dann folgt sehr leicht, daß die Menge aller rekursiv aufzählbaren Mengen aus (I)TG-FIN$_{„\leq 1"}$ ist: Eine Nummer der Sprache wird in diesem (Gegen-)Beispiel kodiert und der Gegenspieler kann keine „Verwirrung" mehr stiften; schließlich wurde das einzige verfügbare Beispiel schon belegt. Es ist nur im Hypothesenraum dafür zu sorgen, daß die entsprechenden Wörter (nicht) in der Sprache liegen.

Eine weitere interessante Frage werfen die Sätze 2.22 und 4.16/17 auf: „Gute Beispiele für eine Sprache...". Wie ist dieses „gut" zu verstehen? Satz 2.22 zeigt, daß es notwendig sein kann, für eine Sprache mehrere Mengen guter Beispiele zur Verfügung zu stellen. Aus 4.17 folgt, daß zwei Schüler unter gewissen Umständen gleiche Beispiele verschieden auswerten. D.h. für beide sind die Beispiele „gut" – sie lernen daraus eine Sprache, aber sie bedeuten für jeden etwas anderes. In [LNW] wurde gesagt, die guten Beispiele seien „typische Informationen" über ein zu erlernendes Objekt. Die hier erhaltenen Resultate legen den Schluß nahe, daß „gute" Beispiele mindestens ebenso „typisch" für den Hypothesenraum sind, wie für die Sprachen, die sie erklären sollen.

• • •

Bei Professor Wiehagen, Werner Stein und Silvia Zilles möchte ich mich herzlich für ihre Hilfe bedanken.

Literaturverzeichnis

[Blum] : Blum, M. (1967), A machine-independent theory of the complexity of recursive functions. *Journal of the Association for Computing Machinery* 14, 322-336.

[Case] : Case, J. (1994), Infinitary self-reference in learning theory. *Journal of Experimental & Theoretical Artificial Intelligence* 6, 3-16.

[FKW a] : Freivalds, R., Kinber, E.B., Wiehagen, R. (1989), Inductive inference from good examples. In: Proceedings, International Workshop on Analogical and Inductive Inference, *Lecture Notes in Artificial Intelligence* 397, 1 - 17.

[FKW b] : Freivalds, R., Kinber, E.B., Wiehagen, R. (1993), On the power of inductive inference from good examples. *Theoretical Computer Science* 110, 131 - 144.

[Friedberg] : Friedberg, R. M. (1958), Three theorems on recursive enumeration. *Journal of Symbolic Logic* 23, 309 - 316.

[Gold] : Gold, E.M. (1967), Language indentification in the limit. *Information and Control* 10, 447 - 474.

[Lachlan] : Lachlan, A.H. (1965), On recursive enumeration without repetition. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 11, 209 - 220.

[LNW] : Lange, S., Nessel, J., Wiehagen, R. (1994), Language learning from good examples. In: Proceedings, International Workshop on Algorithmic Learning Theory, *Lecture Notes in Aritificial Intelligence* 872, 423 - 437.

[LZ a] : Lange, S., Zeugmann, T. (1992), Types of monotonic language learning and their characterisation. In: Proceedings, 5th Annual ACM Workshop on Computational Learning Theory, *ACM Press*, 377 - 390.

[LZ b] : Lange, S., Zeugmann, T. (1993), Language learning in dependence of the space of hypotheses. In: Proceedings, 6th Annual ACM Workhop on Computational Learning Theory, *ACM Press*, 127 - 136.

[Kummer] : Kummer, M. (1989), Numberings of $R_1 \cup F$. In: Proceedings, Computer Science Logic, *Lecture Notes in Computer Science* 385, 166 - 186.

[OSW] : Osherson, D., Stob, M., Weinstein, S. (1986), *Systems that learn*, MIT Press, Cambridge, Massachusetts.

[PP] : Pour-El, M.B., Putnam, H. (1965), Recursively enumerable classes and their application to sequences of formal theories. In: *Archiv für math. Logik und Grundlagenforschung* 8, 104 - 121.

[Rogers] : Rogers, H.Jr. (1987), *Theory of recursive functions and effective computability*, MIT Press, Cambridge, Massachusetts.