

FORSCHUNGSBERICHT
aus dem
INSTITUT FÜR PSYCHOLOGIE

93/1

Determinanten des Erwerbs von
'Programmierkompetenz'

Ute Schmid und Daniela Ulber

93/1

Determinanten des Erwerbs von Programmierkompetenz*

Ute Schmid und Daniela Ulber

Die Arbeit basiert auf einer Untersuchung, die im Rahmen eines Orientierungsprojektes der Allgemeinen Psychologie durchgeführt wurde, das im Sommersemester 1991 und Wintersemester 1991/1992 stattfand. Teilnehmer dieses Projekts waren Katharina Albrecht, Julia Bleser, Werner Breetzke, Stefan Chowdhury, Jens Gräbener, Natalie Juranek, Antje Kuchenbuch, Magdalena Maas, Rainer Marschall, Elisabeth Matthiae, Volker Strenger, Marina Tilgner und Birgit Volbach.

Wir danken den Projektteilnehmern für ihren Einsatz bei der Vorbereitung, Durchführung und Auswertung der Untersuchung. Außerdem danken wir Herrn cand. inform. Guido Dunker für die Unterstützung bei der Konzeption des verwendeten Lernumgebung und Herrn Dr. Uwe Konerding für das Programm zur Raschskalierung. Bei Herrn Prof. Dr. Klaus Eyferth und Herrn Dipl. Psych. Harald Kolrep bedanken wir uns für Anregungen in mehreren Diskussionen.

Nicht zuletzt danken wir allen Probanden, die an der - sicherlich anstrengenden - Untersuchung teilgenommen haben.

Zusammenfassung

Es wird eine empirische Untersuchung vorgestellt, bei der der Einfluß von 1) der Art der vorgegebenen Lernhilfe und 2) dem kognitiven Stil von Personen auf das Lösen von rekursiven Programmieraufgaben und den Lerntransfer geprüft wird. Als Lernhilfen werden dabei erklärende Texte, Beispielprogramme oder beide genannten Materialien verwendet. Zudem wurden die untersuchten Personen aufgrund der Differenz zweier Testskalen als induktive und deduktive Denker klassifiziert.

Es zeigte sich, daß - abweichend von den formulierten Hypothesen - die Aufgabenlösungen besser durch Erklärungen als durch Beispiele unterstützt werden. Dieser Unterschied resultiert jedoch nicht in Unterschieden bezüglich dem Lerntransfer. Bezüglich des Einflusses der Denkstile ergab sich ebenfalls ein nicht den Hypothesen entsprechendes Ergebnis: Personen mit induktivem Denkstil kommen besser mit Erklärungen, Personen mit deduktivem Stil besser mit Beispielen zurecht.

Um genauere Hinweise über den Einfluß verschiedener Lernhilfen auf Aufgabenlösung und Lerntransfer zu erhalten, werden zusätzlich verschiedene Kennwerte zur Erfassung des Hilfeaufrufverhaltens betrachtet.

1 Einleitung

Die Anzahl von Arbeiten, die Bedingungen des effektiven Erwerbs von Programmierkompetenz untersuchen, nimmt ständig zu (z.B. Mayer, 1988; Soloway & Spohrer, 1989). Ein häufig untersuchter Aspekt der Programmierfähigkeit ist das Verstehen und Programmieren rekursiver Kontrollstrukturen. Alle Arbeiten zur rekursiven Programmierung zeigen, daß gerade der Erwerb dieser Programmiermethode besondere Schwierigkeiten bereitet (Kahney, 1989; Kessler & Anderson, 1986, Schömann, 1991). Die Möglichkeit, Personen Rekursion in relativ kurzer Zeit zu vermitteln, ist wiederholt bezweifelt worden.

Ziel der vorliegenden Arbeit ist es, Bedingungen des Erwerbs rekursiver Programmiermethoden zu untersuchen. Als mögliche Einflußfaktoren auf den Lernerfolg werden dabei erstens die Lehrmethode und zweitens der kognitive Stil der Lernenden betrachtet.

Während im Bereich mathematisches Problemlösen eine Vielzahl von Untersuchungen zum Einfluß des Lehrmaterials auf den Wissenserwerb durchgeführt wurden (z.B. Novick & Holyoak, 1991), sind solche Untersuchungen im Bereich Programmierkompetenz selten. In einer Untersuchung von Schmalhofer, Boschert und Kühn (1990) wird der Einfluß von Erklärungen und Beispielen auf den Erwerb von LISP-Syntax verglichen. Diese Arbeit zeigt, daß Beispiele und Erklärungen zu vergleichbaren Transferleistungen führen, also daß diese Materialien informationsäquivalent (Larkin & Simon, 1987) sind. Schmalhofer et al. untersuchten jedoch lediglich den Einfluß von Lehrmaterialien auf den Erwerb der Syntax einer Programmiersprache und beschränkten sich damit auf die deklarativen Anteile des Programmierwissens. Die Beherrschung rekursiver Programmiermethoden ist dagegen eine prozedurale Fertigkeit (vgl. Anderson, 1983).

Wir gehen von der Annahme aus, daß beim Erwerb deklarativen und prozeduralen Wissens zum Teil unterschiedliche Prozesse beteiligt sind. Entsprechend nehmen wir an, daß Beispiele und Erklärungen beim Lernen von Rekursion unterschiedliche Funktionen ausüben und somit die Art und Effizienz des Lernens verschieden beeinflussen.

Als Beispiele werden rekursive Funktionen vorgegeben, die den zu lösenden Programmieraufgaben ähnlich sind, als Erklärungen werden Texte vorgegeben, die die Struktur der zu lösenden Aufgaben abstrakt beschreiben. Programmieranfänger erhalten entweder Beispiele oder Erklärungen oder beide Lehrmaterialien zur Unterstützung. Dabei gehen wir

von folgenden Annahmen aus: Probanden, die Beispiele als Lehrmaterial erhalten, sollten Programmieraufgaben schneller korrekt lösen können, als Probanden, die mit Erklärungen arbeiten, da die Beispielprogramme schablonenhaft auf die zu lösenden Aufgaben übertragen werden können, während abstrakte Erklärungen erst verstanden werden müssen und die Umsetzung der Aufgabenstellung in die Syntax der Programmiersprache selbst geleistet werden muß. Entsprechend erfordert das Arbeiten mit Erklärungen unserer Ansicht eine tiefere Verarbeitung von Information (Craik & Lockhard, 1972) als das Arbeiten mit Beispielen. Dies sollte dazu führen, daß ein Transfer des Gelernten auf neue Aufgabenstellungen und die Beantwortung von Verständnisfragen den Personen, die Erklärungen als Lehrmaterial zur Verfügung hatten. Personen, die beide Lehrmaterialien zur Verfügung haben, sollten sowohl bei der Lösung der Übungsaufgaben, als auch bezüglich der Transferleistung den anderen Lerngruppen überlegen sein, da sie Zugriff auf zwei sich ergänzende Informationstypen haben.

Da viele Untersuchungen im Bereich Programmierwissen eine große interindividuelle Varianz der Programmierleistung berichten (Schömann 1991, Widowski, Eyferth & Schmid, 1991), wird in der hier vorgestellten Arbeit die Personenvariable Denkstil als zweite Determinante des Erwerbs von Programmierkompetenz untersucht. Wir vermuten, daß ein Teil der interindividuellen Varianz darauf zurückzuführen ist, ob Personen einen eher deduktiven oder eher induktiven kognitiven Stil bevorzugen (vgl. Brooks, Simutis & O'Neil, 1983). Deduktive Denkprozesse sind solche, bei denen allgemeine Regeln auf spezielle Probleme übertragen werden, induktive Denkprozesse dagegen sind solche, bei denen aus Beispielen auf allgemeine Regeln geschlossen wird. Entsprechend nehmen wir an, daß die Möglichkeit, Beispiele und Erklärungen als Lehrmaterial sinnvoll zu nutzen, von dem bevorzugten Denkstil einer Person abhängig ist. Konkret sollten induktive Denker besser in der Lage sein, Beispiele anzuwenden und über die Beispiele zu generalisieren, während deduktive Denker eher dazu in der Lage sind, Erklärungen auf konkrete Übungsaufgaben anzuwenden und das aus den Erklärungen erworbene Wissen auf neue Probleme zu übertragen. Bei freier Wahlmöglichkeit des Lehrmaterials sollten Erklärungen von Personen mit deduktivem Denkstil und Beispiele von Personen mit induktivem Denkstil bevorzugt werden.

Im folgenden wird eine Studie berichtet, bei der Programmierneulinge über vier Tage im Programmieren rekursiver Funktionen in einer einfachen funktionalen Sprache unterrichtet wurden. Der Lernprozeß wurde dabei durch die oben genannten Lehrmaterialien unterstützt. Dabei wurde der Einfluß der Lehrmaterialien auf den Lösungserfolg beim Programmieren rekursiver Aufgaben und auf den resultierenden Lerntransfe, sowie die Abhängigkeit der Wirkung der Lehrmaterialien vom ex-post ermittelten kognitiven Stil der Probanden geprüft. Zudem wird das Hilfeaufrufverhalten der Probanden explorativ untersucht.

2 Methode

2.1 Hypothesen und Variablen

Als Determinanten des Erwerbs rekursiver Programmiertechniken werden (1) die Art des Lehrmaterials und (2) der kognitive Stil der Lernenden untersucht. Dabei werden als Lehrmaterial Erklärungstexte (*Erklärungen*), Beispielfunktionen (*Beispiele*) oder beide Materialien (*beides*) vorgegeben. Die untersuchten Probanden werden zusätzlich aufgrund der kritischen Differenzwerte zweier Untertestskaalen des IST-70 (Amthauer, 1970) in eher induktive und eher deduktive Denker unterschieden.

Als abhängige Variablen zur Erfassung des Lernerfolgs werden die Anzahl der gelösten Aufgaben in der Wissenserwerbsphase (*Aufgaben*) und die Transferleistung (*Transfer*) erhoben. Zudem werden als Kennwerte für die Lernstrategie der Probanden Anzahl der Hilfeaufrufe (*Hilfeaufrufe*) und Lesezeiten der Hilfen erfaßt.

Die eingangs formulierten Annahmen zum Erwerb rekursiver Programmiertechniken führten zu folgenden Hypothesen:

- H1 Hauptfaktor: Lehrmaterial (3-fach gestuft)
- H1.1 Aufgaben (beides) > Aufgaben (Beispiele) > Aufgaben (Erklärungen)
 - H1.2 Transfer (beides) > Transfer (Erklärungen) > Transfer (Beispiele)
- H2 Interaktion: Lehrmaterial x Denkstil (2-fach gestuft)
- H2.1 Aufgaben (Erklärungen x deduktiver Stil) >
 - H2.2 Aufgaben (Erklärungen x induktiver Stil)
 - H2.3 Transfer (Erklärungen x deduktiver Stil) >
 - H2.4 Transfer (Erklärungen x induktiver Stil)
 - H2.5 Aufgaben (Beispiele x induktiver Stil) >
 - H2.6 Aufgaben (Beispiele x deduktiver Stil)

- H2.7 Transfer (Beispiele x induktiver Stil) >
- H2.8 Transfer (Beispiele x deduktiver Stil)
- H3 Abhängigkeit der Präferenz von Lehrmaterial in Abhängigkeit vom Denkstil (bei Vorgabe von Erklärungen und Beispielen):
 - H3.1 Hilfeaufrufe (beides x deduktiver Stil): Erklärungen > Beispiele
 - H3.2 Hilfeaufrufe (beides x induktiver Stil): Beispiele > Erklärungen.

Zur Kontrolle werden weitere Variablen erhoben, die erstens die Begabung zum mathematisch-formalen Denken und zweitens die Beherrschung der Syntax der in der Untersuchung verwendeten Programmiersprache erfassen.

2.2 Probanden

An der Untersuchung nahmen 49 Studenten teil, die sich im Grundstudium der Psychologie befanden. Die Probanden wurden durch Zufall auf eine der drei Lernbedingungen verteilt: 17 Probanden erhielten Erklärungen und je 16 Probanden Beispiele oder beide Materialien als Lehrmaterial. Die Einteilung der Probanden nach ihrem kognitiven Stil führt bei den Lernbedingungen "Erklärungen" (deduktiv=4, induktiv=5, nicht klassifizierbar=8 Probanden) und "Beispiele" (deduktiv=4, induktiv=5, nicht klassifizierbar=7 Probanden) zu für statistische Analysen akzeptable Zellbesetzungen. Für die Lernbedingung "Erklärungen und Beispiele" ergaben sich dagegen leider sehr ungünstige Zellbesetzungen (deduktiv=1, induktiv=7, nicht klassifizierbar=8 Probanden). Damit ist die dritte Hypothese (siehe 2.1, H3) über die Abhängigkeit der Präferenz von Lernhilfen vom Denkstil nicht statistisch prüfbar.

2.3 Lerninhalt

Den Probanden wurde eine selbstentwickelte funktionale Programmiersprache mit sehr geringem, schnell erlernbarem Syntaxumfang vermittelt. Funktionale Sprachen betrachten Programme als Funktionen von Mengen von Eingabewerten in Mengen von Ausgabewerten (vgl. z.B. Field & Harrison, 1988). Die klassische Kontrollstruktur funktionaler Sprachen ist die Rekursion. Rekursive Programme enthalten im Rumpf einen Aufruf ihrer selbst. Sie entsprechen in etwa rekursiven Definitionen, wie sie in der Mathematik verwendet werden, bei denen eine Funktion "durch sich selbst" definiert wird. So ist zum Beispiel die Fakultät

(n!) folgendermaßen definiert:

$$n! = \begin{cases} 1 & \text{wenn } n \leq 1 \\ n * (n-1)! & \text{sonst} \end{cases}$$

2.4 Lernumgebung und Instruktionen

Auf den Interpreter der funktionalen Sprache wurde eine Lernumgebung (LEAR) aufgesetzt (siehe Abb. 1). Der Bildschirm von LEAR ist in ein Editor-, ein Interpreter- und ein Kommentarfenster unterteilt. Im Editor erhält der Lernende die Aufgabenstellung und kann dort einen Funktionscode erstellen. LEAR prüft die Aufgabe auf syntaktische und semantische Korrektheit¹ und meldet diese Fehler zurück. Eine korrekte Funktion kann im Interpreterfenster mit konkreten Eingabewerten ausgeführt werden. Nach 30 Minuten oder nach drei semantisch fehlerhaften Lösungsversuchen wird die richtige Lösung bekanntgegeben.

¹semantische Korrektheit wird durch ein internes interpretieren der Aufgabenlösung mit kritischen Testwerten geprüft. Dabei sind auch nicht-terminierende Rekursionen erkennbar.

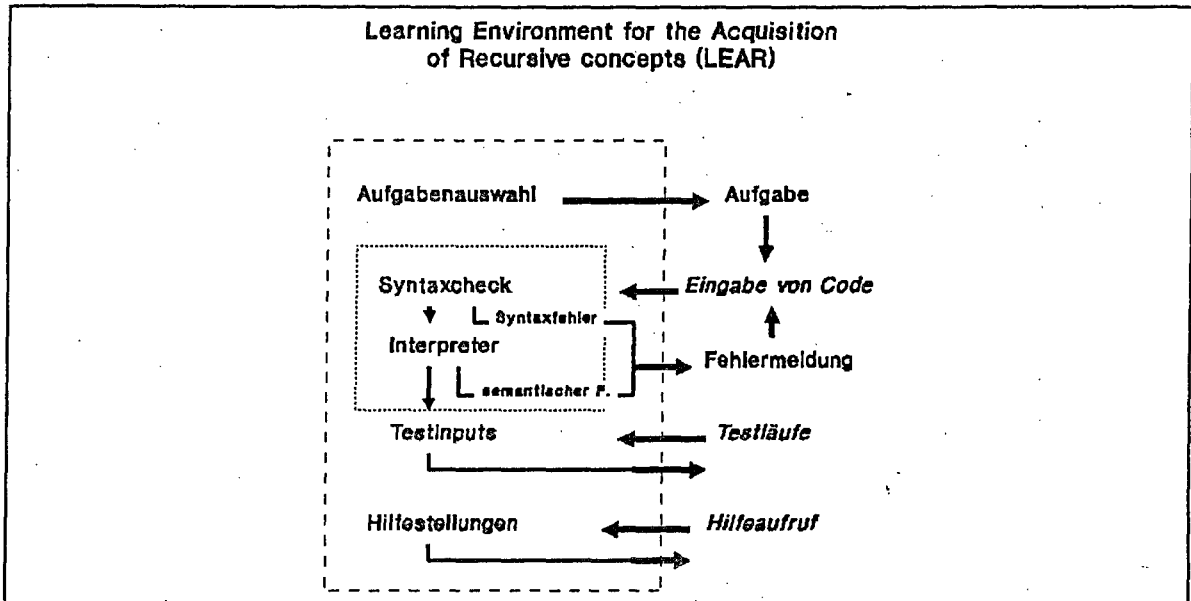


Abb. 1: Die Struktur von LEAR

Für jede rekursive Aufgabenstellung können zwei Hilfstexte (Erklärungen, Beispiele oder beides) beliebig oft aufgerufen werden. Um die Häufigkeiten der Hilfeaufrufe, sowie die Lesezeiten erfassen zu können, muß die Hilfe jeweils ausgeschaltet werden, bevor an der Aufgabenlösung weiter gearbeitet werden kann.

2.5 Testmaterial und Fragebögen

In der Untersuchung wurden folgende Fragebögen und Tests verwendet:

Demographischer Fragebogen

Neben allgemeinen Daten wie Alter, Geschlecht und Semesteranzahl wurden auch Leistungskurse, letzte Mathematiknote sowie Schreibmaschinen- und Englischkenntnisse

der Probanden erfragt.

- Amthauer Intelligenz-Strukturtest (IST-70)

Zur Erfassung des bevorzugten kognitiven Stils der Probanden wurden die Untertests 5 (Rechenaufgaben) und 6 (Zahlenreihen) des IST-70 verwendet. Nach Amthauer (1970, S.39) erfaßt der Untertest 5 praktisch-rechnerisches, deduktives, der Untertest 6 theoretisch-rechnerisches, induktives Denken. Aufgrund der kritischen Differenzwerte zwischen den beiden Untertests (Lienert, 1961; S. 458ff.) wurden die Probanden als induktive bzw. deduktive Denker klassifiziert.

- Syntaxtest

Um zu kontrollieren, inwieweit die Probanden über ausreichende bzw. vergleichbare Kenntnisse über die Syntax der verwendeten Programmiersprache verfügten, wurde ein aus 28 Einzelitems bestehender Syntaxtest konstruiert. den die Probanden nach Abschluß der Syntaxvermittlung bearbeiteten.

- Abschlußtest

Zur Erfassung des Lerntransfers der Probanden wurde ein aus 21 Items bestehender Abschlußtest entwickelt. Die Items erfassen vier verschiedene Aspekte des erworbenen Wissens über Rekursion: Wissen über Merkmale rekursiver Funktionen, Verständnis der Semantik von vorgegebenen Funktionen, Ein-Ausgabe-Verhalten von Funktionen und Programmierung rekursiver Funktionen. Als abhängige Variable wurden die geschätzten Fähigkeitsparameter des raschskalierten Abschlußtests verwendet (vgl. Sixtl, 1982).

- Evaluationsfragebogen

Schließlich wurden die Probanden über ihre Zufriedenheit mit dem Experiment und die Selbsteinschätzung ihres Lernerfolges befragt. Zudem wurde Information über die subjektive Beurteilung der Lernhilfen erhoben. Zum einen sollte der Unterstützungswert der Lernhilfen auf einer 7-stufigen Ratingskala eingeschätzt werden, zum anderen sollten die Probanden angeben, in welchen Situationen ihnen die Lernhilfe besonders nützlich war.

2.6 Untersuchungsablauf

Die Untersuchung fand vom 14.10.91 bis zum 07.11.91 im PC-Pool des Institutes für Psychologie der Technischen Universität Berlin statt. Der Kapazität des PC-Pools entsprechend wurden 6 Gruppen gebildet. Jede Gruppe nahm über die Dauer einer Woche von Montag bis Donnerstag an insgesamt 4 Sitzungen von 15-18 Uhr bzw von 18-21 Uhr teil. Die Untersuchungsteilnehmer wurden durch Losverfahren einer der 3 Versuchsbedingungen zugeordnet. In den ersten zwei Sitzungen wurde ausschließlich die Syntax der Programmiersprache vermittelt. Diese Einführung erfolgte durch Frontalunterricht in Kombination mit interaktiven Übungen am Rechner. Die in diesem Rahmen bearbeiteten Übungsaufgaben gehen nicht in die Auswertung mit ein. Während der folgenden Sitzungen, in denen die Rekursion gelernt wurde, verfügten die Lernenden über ein Merkblatt mit der gesamten Syntax. Tabelle 1 gibt einen Überblick über die Untersuchungsdurchführung.

Tab. 1: Zeitablauf der Untersuchung

1. Sitzung (Mo)

* Einführung: Überblick über die Untersuchung

*** ERHEBUNG VON PERSONENDATEN ***

* Demographischer Fragebogen: demographische Daten und Mathematikkenntnisse

* IST-70 Rechenaufgaben + Zahlenreihen: Erfassung von Denkstilen

*** EINFÜHRUNGSPHASE ***

* Pause (10 Min)

* Lektion 1

Syntax 1: Definition von Funktionen, Datentypen, arithmetische Anweisungen

Interpretation von Funktionen

Hinweise zur PC-Bedienung

*** am PC: (1 Stunde)

- interaktive Einführung in LEAR

- Aufgaben 1 und 2

2. Sitzung (Di)

* Wiederholung von Syntax 1

* Lektion 2

Syntax 2a: boolesche Ausdrücke, bedingte Anweisung

Syntax 2b: Listen

Syntaktische und semantische Korrektheit

* Pause (10 Min)

*** am PC: (2 Stunden)

- interaktive Wiederholung der Syntax

- Aufgaben 3 und 4 zu Syntax 2a

- Aufgaben 5 und 6 zu Syntax 2b

3. Sitzung (Mi)

* Syntaxtest: Kontrolle der Vergleichbarkeit des Syntaxwissens

* Allgemeine Einführung der Rekursion

* Hinweis zum Aufruf der Lernbedingungen

* Pause (10 Min)

*** EXPERIMENTELLE PHASE ***

*** am PC: (1,5 Stunden)

- Aufgaben 7,8,9: Endrekursion

4. Sitzung (Do)

*** am PC: (1,5 Stunden)

- Aufgaben 10,11,12: Teil-Rest-Rekursion

*** TESTPHASE ***

* Pause (10 Min)

* Abschlußtest: Erfassung des Lerntransfers

* Kurzfragebogen zur Evaluation der Untersuchung und der Lernhilfen

* Entlohnung

3 Ergebnisse

Der Lernerfolg der Probanden, die ja fast alle keine Programmierererfahrung hatten, war überraschend hoch: Von den 6 in Lektion 3 und 4 zu bearbeitenden Aufgaben wurden im Mittel 2.57 Aufgaben (sd = 1.90) gelöst, wobei es Probanden gab, die keine Aufgabe bzw. alle 6 Aufgaben lösten.

Von den 21 Items des Abschlußtests wurden im Schnitt 9.71 gelöst (sd = 3.07). Das Maximum betrug 17, das Minimum 4 Items. Alle statistischen Analysen werden nicht mit den Rohwerten, sondern mit den rasch-skalierten Fähigkeitsparametern durchgeführt.

Einen Überblick über die Ausprägungen aller erfaßten Variablen in der Gesamtstichprobe gibt Tabelle 2.

Tab. 2: Ausprägung der Variablen in der Stichprobe

	Mittelwert	Standardabw.	Minimum	Maximum
<u>Kontrollvariable</u>				
Syntaxtest (28 Items)	15.39	5.32	4	26
<u>Abhängige Variablen</u>				
Transferleistung* (21 Items)	9.71(0)	3.07(1.25)	4(-2.94)	17(2.7)
Anzahl gel. Aufgaben (6 Aufgaben)	2.57	1.9	0	6
<u>Begabungsmaße</u>				
Rechenaufgaben	103.41	11.55	79	128
Zahlenreihen	106.41	11.27	82	125
Mathenote	2.80	1.21	1	6
Mathe-Selbsteinsch.	2.46	0.92	1	4
<u>Strategiemaße**</u>				
Häufigkeit der Hilfeaufrufe	45.43	34.41	2	146
Mittlere Lesezeit der Hilfeaufrufe	114.36s	133.43s	23.98s	652.04s

n = 49

* = Raschskalenwerte in Klammern

** Häufigkeit = Summe der Aufrufe über alle 6 verwendeten Aufgaben;
Lesezeit = mittlere Lesezeit pro Hilfeaufruf

3.1 Einfluß der Lernbedingung auf den Lernerfolg

3.1.1 Lerntransfer

Bezüglich des Lerntransfers, erfaßt durch die Fähigkeitsparameter des raschskalierten Abschlußtests, ergeben sich lediglich vernachlässigbare Unterschiede zwischen den drei Lernbedingungen (ANOVA mit Faktor Lernbedingung: $F=0.18$, $df=2/46$, $p=0.83$): Probanden, die Beispiele und Erklärungen zur Verfügung hatten zeigten den höchsten, Probanden, die nur Beispiele zur Verfügung hatten, den geringsten Lernerfolg (siehe Abb. 2).

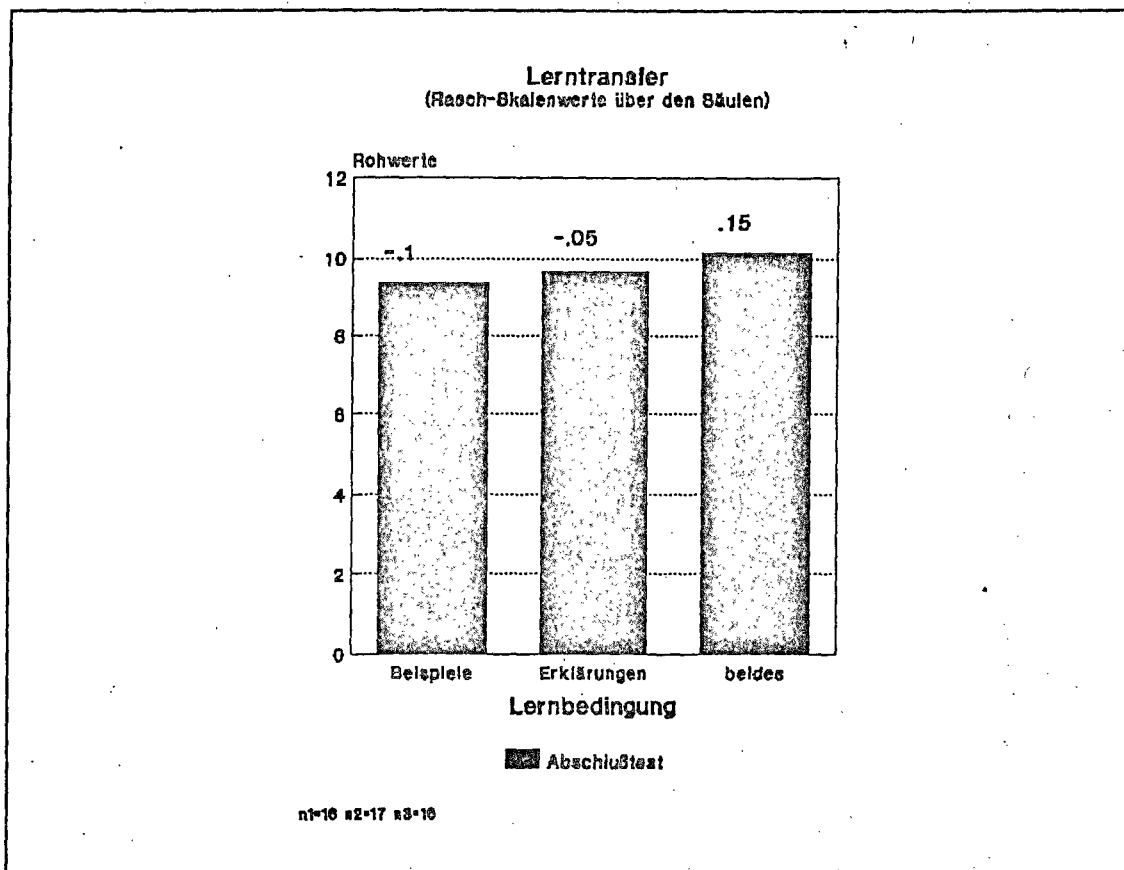


Abb. 2: Einfluß der Lernbedingungen auf die Transferleistung

Dieses Ergebnis kann die eingangs formulierten Hypothese nicht stützen, wonach die Probanden, die mit Beispielen arbeiten, deutlich weniger Lerntransfer haben sollten, als die Probanden, die mit Erklärungen arbeiten.

3.1.2 Leistungen beim Lösen der Aufgaben

Die Probanden, die durch Erklärungen unterstützt wurden, lösten insgesamt 50% der Aufgaben, diejenigen, die durch Beispiele unterstützt wurden; 25% und die Gruppe, die beide Materialien zur Verfügung hatte, 54% der Aufgaben (siehe Abb. 3). Diese Unterschiede zwischen den Bedingungen sind hochsignifikant (Chi-krit = 9.21, Chi-emp = 12.5, $p < .001$).

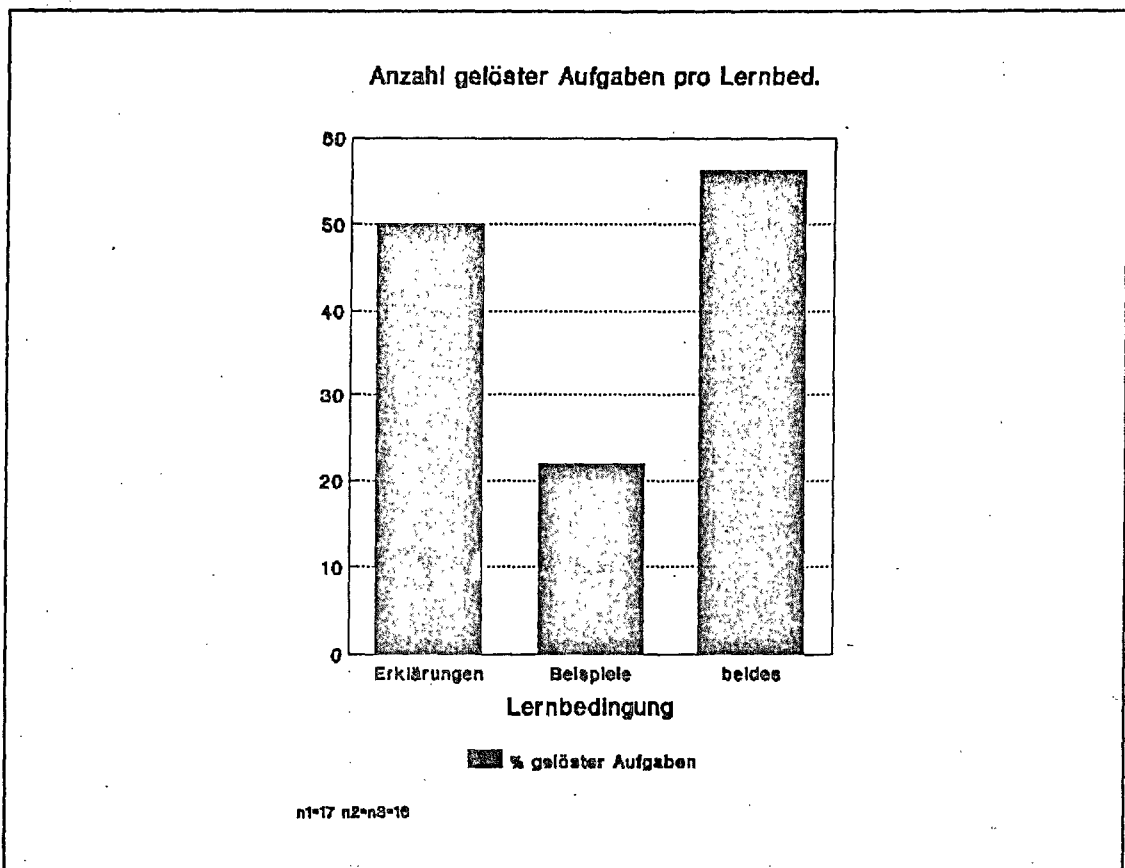


Abb. 3: Einfluß der Lernbedingung auf die Aufgabenlösung

Dieses Ergebnis steht deutlich im Widerspruch zu der Hypothese, daß Probanden, die mit Beispielen arbeiten, deutlich mehr Aufgaben lösen sollten, als Probanden, die mit Erklärungen arbeiten.

Auffällige Unterschiede zeigen sich auch bei der Lösungszeit der Übungsaufgaben: Die Beispielgruppe benötigte für die Lösung der Aufgaben aus Lektion 3 und 4 insgesamt 4.54

Stunden, die Erklärungsgruppe 5.07 Stunden und die Gruppe, die beide Materialien zur Verfügung hatte, 3.91 Stunden. Die Unterschiede werden bei einer einfaktorischen Varianzanalyse auf dem 5% Niveau signifikant ($F=4.251$, $df=2/42$, $p=0.021$; 4 missing values).

Die deutlich längeren Lösungszeiten lassen vermuten, daß die Übertragung von Beispielfunktionen auf Aufgaben aufwendige Verarbeitungsprozesse benötigt und die Anpassung der Beispiele an die Aufgabe nicht ohne ein Verständnis der Beispielfunktionen gelingt.

3.1.3 Zusammenhang zwischen den Leistungen beim Lösen der Übungsaufgaben und beim Transfer-/Verständnistest

Betrachtet man den Zusammenhang von Erfolg bei der Aufgabenlösung und Lerntransfer, so ergeben sich signifikante Zusammenhänge für die Erklärungsgruppe ($r = .74^{**}$) und die Gruppe, die beide Lernmaterialien zur Verfügung hatte ($r = .73^{**}$). Der Zusammenhang zwischen Erfolg bei der Aufgabenlösung und Lerntransfer bei der Beispielgruppe ist dagegen nicht bedeutsam ($r = .22$).

Dieses Ergebnis kann möglicherweise durch die Zeitbegrenzung bei der Aufgabenlösung zustande gekommen sein: Die Beispielgruppe benötigte wesentlich mehr Zeit beim Aufgabenlösen und erreichte dennoch kaum korrekte Lösungen innerhalb der Maximalzeit von 30 Minuten. Dennoch scheint diese Lerngruppe bei dem Versuch, die Aufgaben mithilfe von Beispielen zu lösen, etwas gelernt zu haben, auch wenn die Zeit nicht ausreichte, die Aufgaben vollständig zu bearbeiten.

3.2 Einfluß der Interaktion von Lernbedingung und Denkstil auf den Lernerfolg

Die Probanden wurden ex-post aufgrund der kritischen Differenzwerte bei den IST-Unterskalen nach ihrem bevorzugten Denkstilen in induktive, deduktive und nicht eindeutig klassifizierbare (offene) Denker eingeteilt.

Bezüglich der zweiten betrachteten Einflußgröße auf den Lernerfolg, den ex-post ermittelten Denkstil der Probanden ergibt sich weder bezüglich des Lerntransfers ($F=0.41$, $df=1$, $p=0.532$), noch bezüglich der Anzahl gelöster Aufgaben ($F=0.39$, $df=1$, $p=0.540$) ein signifikanter Haupteffekt.

3.2.1 Interaktion von Lernbedingung und Denkstil bezüglich des Lerntransfers

Betrachtet man die Interaktion von Lernbedingung und Denkstil (siehe Abb. 4), so erreichen bei der Erklärungsbedingung diejenigen Probanden, die keinem Denkstil eindeutig zuordbar sind ("offene Denker") die besten Ergebnisse, gefolgt von den induktiven Denkern. Die deduktiven Denker weisen hier die schlechtesten Ergebnisse auf.

In der Beispielbedingung dagegen erreichen die deduktiven Denker bessere Transferleistungen als die induktiven.

Diese Interaktion zwischen Lernbedingung und Denkstil wird signifikant ($F=5.224$, $df=1$, $p=0.038$), wenn die Lernbedingung, bei der Erklärungen und Beispiele zur Verfügung standen, und die keinem Denkstil zuordbaren Probanden nicht in die Rechnung mit einbezogen werden.

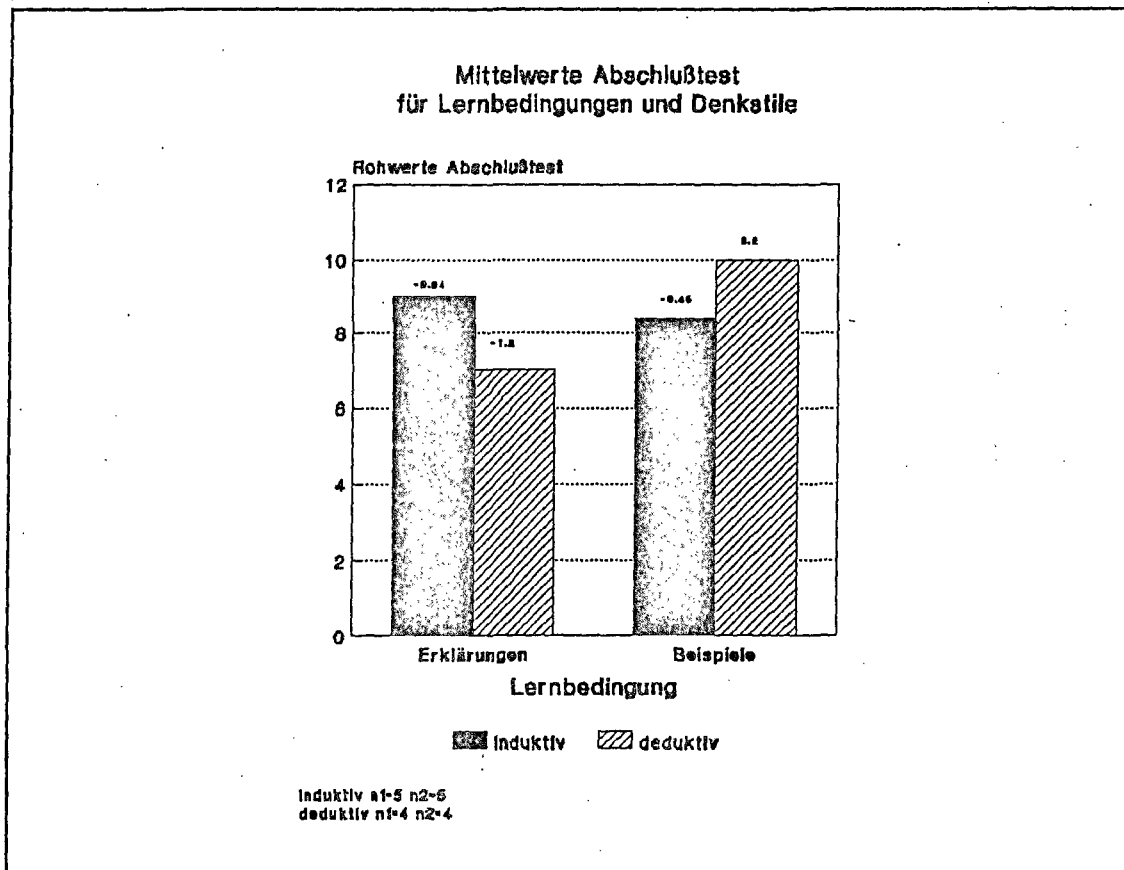


Abb. 4: Einfluß der Interaktion von Lernbedingung und Denkstil auf die Transferleistung

3.2.2 Interaktion bezüglich der Lösungen der Übungsaufgaben

In den Lernbedingungen "Beispiele" und "Erklärungen" werden die meisten Aufgaben von den "offenen Denkern" gelöst. Bei Betrachtung der induktiven und deduktiven Denker zeigt sich die gleiche Tendenz wie beim Abschlußtest: In der Erklärungsbedingung erreichen die induktiven Denker, in der Beispielbedingung die deduktiven Denker die besseren Leistungen.

Diese Interaktion wurde jedoch nicht signifikant ($F=0.147$, $df=1$, $p=0.244$; siehe Abb. 5).

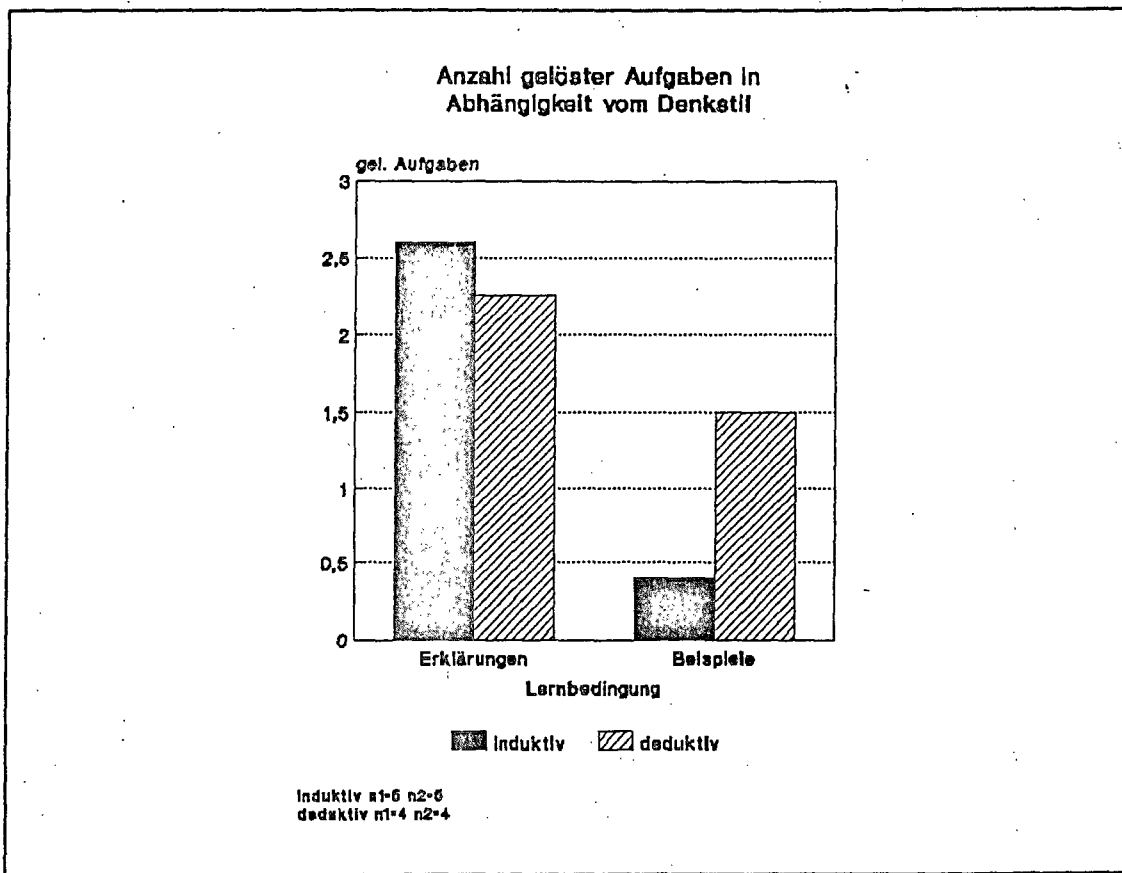


Abb. 5: Einfluß der Interaktion von Lernbedingung und Denkstil auf die Aufgabenlösung

3.3 Häufigkeit und mittlere Lesezeit der Hilfeaufrufe

Um genaueren Aufschluß über den Einfluß der verwendeten Lehrmaterialien zu erhalten, werden im folgenden Häufigkeit der Hilfeaufrufe und Lesezeiten pro Hilfe betrachtet.

Die Häufigkeit von Hilfeaufrufen war unter der Lernbedingung mit Beispielen am höchsten und unter der Lernbedingung mit Erklärungen am niedrigsten. Diese Unterschiede sind jedoch nicht signifikant. Durchschnittlich erfolgten unter der Erklärungsbedingung 36 (min=4, max=66), unter der Beispielbedingung 52 (min=2, max=146) und unter der Bedingung, die beide Materialien zur Verfügung hatte, 48 (min=8, max=123) Hilfeaufrufe (siehe Abb. 6).

Auffällig ist, daß die mittlere Lesezeit für Beispiele doppelt so hoch wie die für Erklärungen ist. Die Probanden benötigten zum Lesen der Erklärungen durchschnittlich 7.5 und zum Lesen der Beispiele 15 Minuten pro Aufgabe.

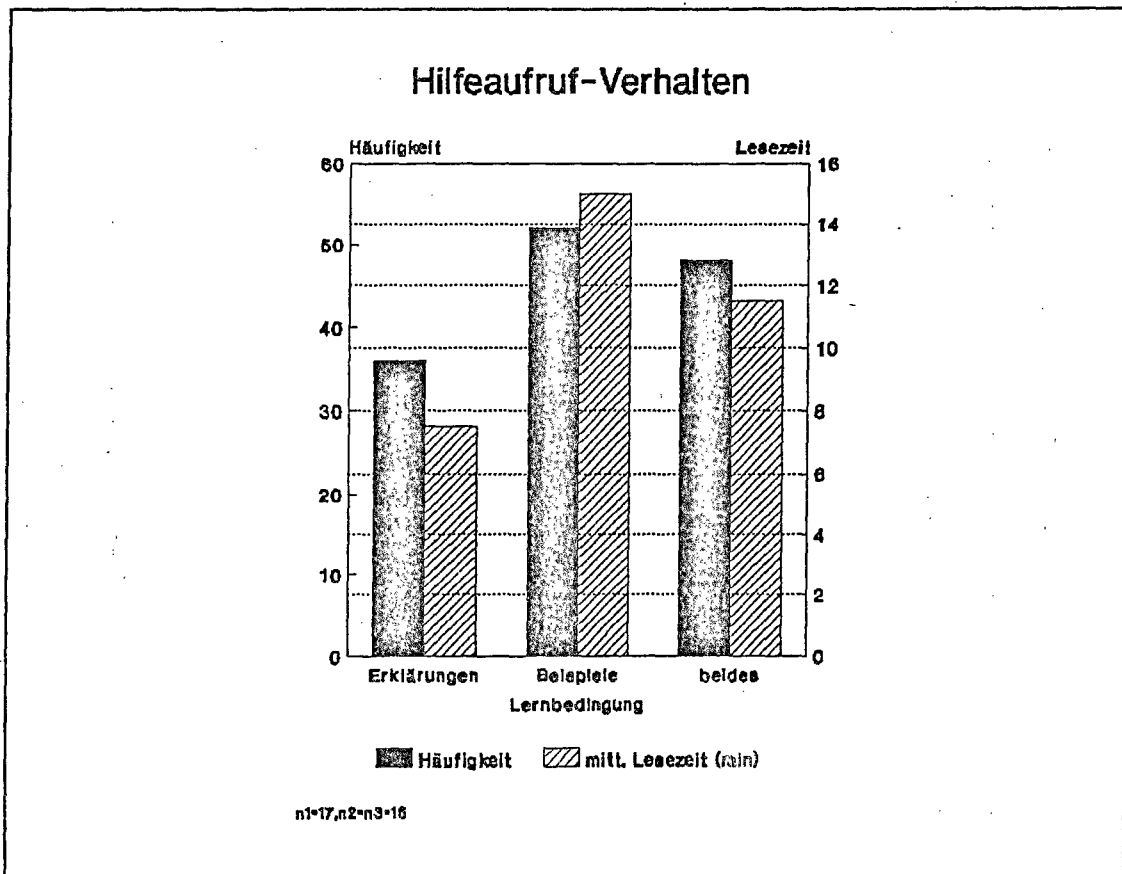


Abb. 6: Häufigkeiten und mittlere Lesezeiten der Hilfeaufrufe

3.4 Präferenz von Lernhilfen bei der Probandengruppe, die Erklärungen und Beispiele zur Verfügung hat

Ein direkter Vergleich der Nutzung der Lernhilfen kann bei den Probanden durchgeführt werden, die sowohl Erklärungen, als auch Beispiele zur Verfügung hatte. Diese Gruppe konnte selbst entscheiden, welche Lernhilfe sie jeweils auswählte.

Bezüglich der Häufigkeiten und der durchschnittlichen Lesezeiten der Hilfeaufsarten ergeben sich keine signifikanten Unterschiede.

3.4.1 Einfluß des kognitiven Stils auf das Hilfeaufverhalten

Aufgrund der ungünstigen Zellenbesetzung für die ex-post Variable Denkstil (siehe Punkt 2.2) lassen sich für die deduktiven Denker (nur 1 Proband in der Lernbedingung "Erklärungen und Beispiele") keine statistischen Aussagen über den Einfluß des kognitiven Stils auf das Hilfeaufverhalten machen.

Betrachtet man die Anzahl und die Lesezeit der Erklärungs- und Beispielaufrufe, so zeigt sich, daß alle drei Denkstilgruppen Erklärungen häufiger als Beispiele aufrufen und die Beispiele länger lesen. Interessanterweise sind diese Differenzen bei der Gruppe der induktiven Denker am größten, d. h. gerade die induktiven Denker präferieren die Erklärungen am meisten und lesen die Beispiele am längsten (siehe Abb. 7). Allerdings sind weder die Unterschiede der Denkstilgruppen bezüglich der Aufrufhäufigkeit von Erklärungen ($F=0.395$, $df=2$, $p=0.681$) und Beispielen ($F=0.359$, $df=2$, $p=0.705$) noch die Differenzen der Aufrufhäufigkeit und der Lesezeit bei den induktiven Denkern ($T=1.754$, $df=6$, $p=0.13$ für die Aufrufhäufigkeit und $T=-0.507$, $df=6$, $p=0.63$ für die Lesezeit) signifikant. Der Trend bei den induktiven Denkern widerspricht jedoch unserer Hypothese, nach der induktive Denker Beispiele bevorzugen sollten.

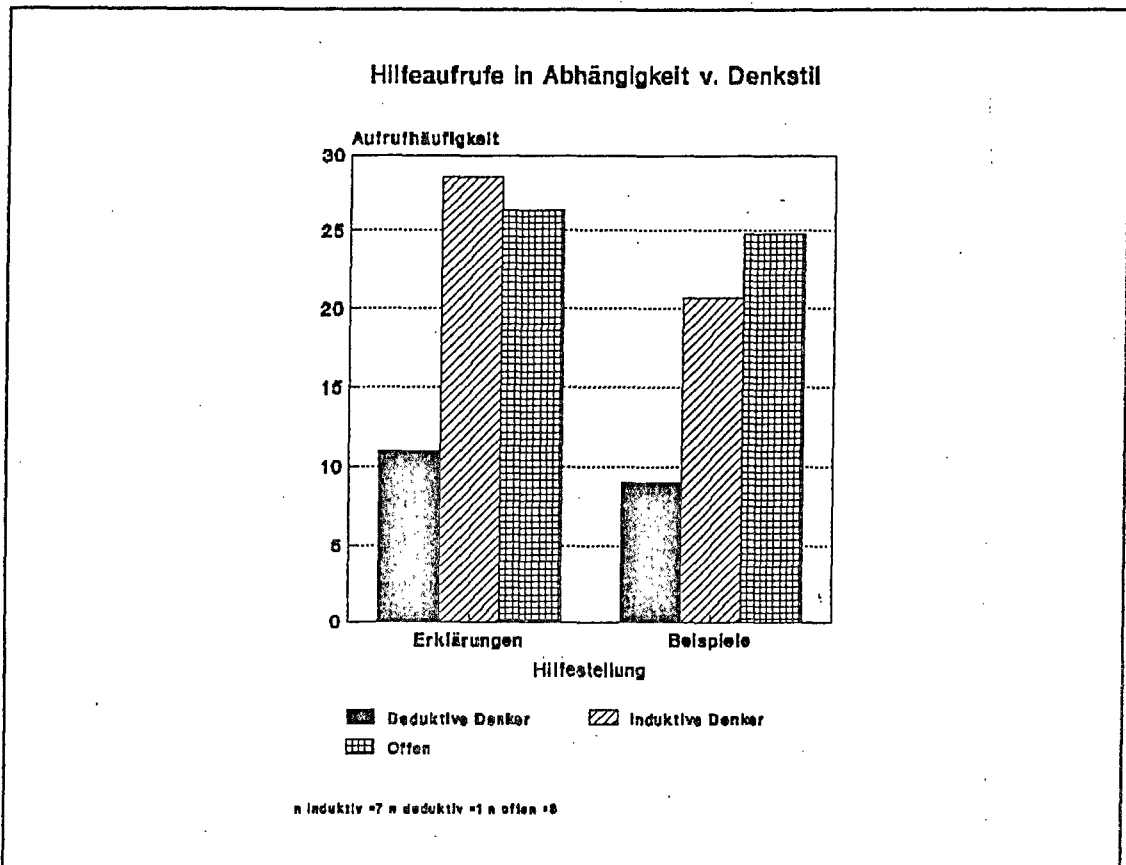


Abb. 7: Hilfeaufrufverhalten in der Lernbedingung "beides"

3.4.2 Subjektive Einschätzung der Unterstützung durch Erklärungen und Beispiele

Betrachtet man die subjektiven Einschätzungen des Nutzens der Lernhilfen aus dem Evaluationsfragebogen, zeigt sich jedoch, daß die Probanden, die beide Materialien zur Verfügung hatten, Erklärungen und Beispiele zwar in gleichem Maße für hilfreich halten, sie aber dennoch unterschiedlich bewerten.

Die Frage, ob sie mit Erklärungen als alleinige Hilfe gut zurechtgekommen wären bejahten 5 Probanden. Dagegen glaubten nur 2 Probanden, daß ihnen Beispiele genügt hätten. 8 Probanden waren nicht sicher, ob eine der Bedingungen für sie ausgereicht hätte.

Auf die offene Frage, wann die Erklärungen, wann die Beispiele mehr geholfen hätten, antworteten die Probanden wie folgt (siehe auch Tab. 3).

7 Probanden bevorzugten Erklärungen, wenn sie Schwierigkeiten mit der Semantik hatten, dagegen bevorzugten nur 2 Probanden Erklärungen bei Syntaxfragen. Beispiele wurden von

7 Probanden bei Syntaxfragen und nur von einem Probanden bei Semantikproblemen bevorzugt.

3 Probanden riefen besonders am Anfang Erklärungen ab und keiner am Ende. Beispiele wurden genau umgekehrt abgerufen. Dieses Abrufverhalten ist als deutlich hypothesenkonform zu bezeichnen.

Tab. 3: Evaluation der Unterstützung durch die Lernhilfen bei der Kontrollgruppe

	Semantik	Syntax	Anfang	Ende
Erklärungen	7	2	3	0
Beispiele	1	7	0	3

4 Diskussion

Unsere beiden zentralen Hypothesen, daß Beispiele das Lösen von Aufgaben besser unterstützen und Erklärungen zu besserem Lerntransfer führen, konnten nicht bestätigt werden.

Der geringe Unterschied beim Lerntransfer zwischen den drei Bedingungen entspricht den Befunden von Schmalhofer et al. (1989), wonach Beispiele und Erklärungen informationsäquivalente Materialien sind. Der - für uns überraschende - Vorsprung der Erklärungsgruppe bei der Aufgabenlösung legt die Vermutung nahe, daß die Übertragung von Beispielen auf zu lösende Programmieraufgaben komplexe Verarbeitungsprozesse und ein volles Verständnis der Beispielfunktionen voraussetzt (vgl. auch Lefrancois, Kap.10, 1976). Die erklärenden Texte scheinen dagegen recht einfach auf die Aufgabenlösung anwendbar zu sein. Es wäre möglich, daß der Zwang, die Lernhilfen auszublenden, bevor weiter an der Aufgabenlösung gearbeitet werden kann, den deutlichen Nachteil für die Beispielgruppe stark beeinflusst hat: Natürlichsprachig formulierte Erklärungen können wahrscheinlich besser im Gedächtnis behalten werden als die in der noch unvertrauten Programmiersprache formulierten Beispielfunktionen.

Die Probleme, Beispiele zur Aufgabenlösung einzusetzen, werden durch das beobachtete Hilfeaufrufverhalten der Probanden unterstützt. Hier hat sich gezeigt, daß Erklärungen kürzer gelesen werden als Beispiele.

In der Bedingung, bei der beide Lehrmaterialien zur Verfügung standen, werden Erklärungen in der Tendenz kürzer gelesen und häufiger aufgerufen als Beispiele. Bei der Frage nach dem Nutzen der beiden Lehrmaterialien werden von den meisten Probanden beide Hilfestellungen für ihren Lernerfolg für notwendig erachtet und als gleich hilfreich angesehen.

Interessant ist es, daß sich der Denkstil tatsächlich auf die Lernleistung, insbesondere auf die Transferleistung je nach Lernbedingung auswirkt - wenn auch nicht unseren Hypothesen entsprechend. Die Tatsache, ob man einen eher induktiven oder eher deduktiven Denkstil hat, erklärt demnach tatsächlich einen Teil der Varianz der Leistungen beim Programmierenlernen. Bemerkenswert erscheint auch der Trend, daß die Probanden, deren IST-Teiltestergebnisse zu ähnlich für eine eindeutige Zuordnung zu einem Denkstil sind, beim Abschlußtest und bezüglich der Zahl gelöster Aufgaben am erfolgreichsten sind. Handelt es sich hier auch nur um eine Tendenz, da die Leistungsunterschiede zu den anderen Denkstilen nicht signifikant sind und die Trennung von den beiden Denkstilen nur ungenau ist (die Gruppe der "offenen Denker" wurde ja gerade zur eindeutigen Abgrenzung von den beiden anderen Gruppen geschaffen), könnte man es doch als Anzeichen dafür werten, daß eine etwa gleich gute Ausprägung induktiven und deduktiven Denkvermögens und eine damit verbundene flexible Anwendung beider Denkstile sich positiv auf die Lernleistung auswirken.

Betrachtet man die besseren Leistungen bei Transfer und Aufgabenlösung der induktiven Denker in der Erklärungsbedingung und der deduktiven Denker in der Beispielbedingung sowie den Trend der induktiven Denker der Kontrollbedingung, mehr Erklärungen aufzurufen und Beispiele länger zu lesen, so muß man zu einem unserer Hypothese entgegengesetzten Schluß gelangen: Personen scheinen jeweils durch das Lehrmaterial besser unterstützt zu werden, das ihrem dominanten Denkstil entgegengesetzt ist.

Positiv anzumerken bleibt, daß die verwendete Lernumgebung LEAR erfolgreich dazu eingesetzt werden konnte, Programmieranfängern in kurzer Zeit Grundlagen des rekursiven Programmierens beizubringen.

Literatur

- Anderson, J. R. (1983). *The architecture of cognition* (Cognitive Science Series 5). Cambridge, MA: Harvard University Press.
- Amthauer (1970). *Intelligenz-Struktur-Test (IST-70). Handanweisung für die Durchführung und Auswertung*. Göttingen: Hogrefe.
- Brooks, L.W., Simutis, Z.M. & O'Neil, H.F. (1983). The role of individual differences in learning strategies research. In R.F. Dillon and R.R. Schmeck (Ed.), *Individual Differences in Cognition* (Vol. 2, pp. 219-251). New York: Academic Press.
- Kahney, H. (1989). What do novice programmers know about recursion? In E. Soloway a. J.C. Spohrer (Ed.), *Studying the Novice Programmer* (pp. 209-228) Lawrence Erlbaum.
- Kessler, C.M. & Anderson, J.R. (1986). Learning flow of control: recursive and iterative procedures. *Human-Computer-Interaction*, 2, 135-166.
- Larkin, J.H. & Simon, H.A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65-99.
- Mayer, R.E. (1988). *Teaching and Learning Computer Programming: Multiple Research Perspectives*. Erlbaum.
- Novick, L.R. & Holyoak, K.J. (1991). Mathematical problem solving by analogy. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17 (3), 398-415.
- Schömann, M. (1991). Erwerb von Rekursionsschemata in LISP. Lernprozesse, Lösungsstrategien und Wissensorganisation von AnfängerInnen und fortgeschrittenen ProgrammiererInnen. Unveröffentlichte Dissertation. TU Braunschweig.
- Soloway, E. & Spohrer, J.C. (Eds.). (1989). *Studying the Novice Programmer*. Lawrence Erlbaum.
- Widowski, D., Eyferth, K. & Schmid, U. (1991). Kognitive Prozesse beim Lesen und Verstehen von Computerprogrammen. *Forschungsinitiativprojekt 16 der TU Berlin, Abschlußbericht*.

Anschrift der Autoren

Ute Schmid
Institut für Psychologie
der TU Berlin
Dovestr. 1-5
1000 Berlin 10

e-mail: utesch@cs.tu-berlin.de