DOI: 10.20378/irb-90971

Erwerb rekursiver Programmiertechniken als Induktion von Konzepten und Regeln

Ute Schmid und Barbara Kaup

Institut für Psychologie der TU Berlin, Dovestr. 1-5, 10587 Berlin

Der induktive Erwerb von Problemlösefertigkeiten wird häufig im Rahmen des analogen Problemlösens betrachtet. Beim analogen Problemlösen lassen sich vier Teilprozesse unterscheiden (vgl. Novick & Holyoak in [3]): (1) der Abruf einer dem Zielproblem ähnlichen Lösung (*retrieval*), (2) der Vergleich von Relationen im Beispielproblem mit denen im Zielproblem (*mapping*), (3) die Anpassung des Lösungsweges im Beispielproblem an das Zielproblem unter Berücksichtigung der Ergebnisse des Vergleichsprozesses (*adaptation*) und (4) der Aufbau eines abstrakten Schemas für die Problemklasse, die durch Ziel- und Beispielproblem repräsentiert wird (*schema induction*). Es wird angenommen, daß induzierte Schemata nachfolgende Problemlösungen erleichtern (Anderson & Thompson in [1]; Carbonell in [2]).

Die für das analoge Problemlösen zentrale Annahme, daß abstraktes Wissen über eine Problemklasse inferiert wird, kann auf Problembereiche übertragen werden, die durch mehrere strukturell unterschiedliche Problemklassen beschreibbar sind. In diesem Fall muß zusätzlich zu den inferierten Schemata eine Organisationsstruktur für diese Schemata erschlossen werden. Eine mögliche Organisationsstruktur für Problemklassen ist eine Hierarchie von Schemata (vgl. Vorberg & Goebel in [5]).

Aus Einzelfallstudien zum Erwerb von Programmierfertigkeiten in LISP ergeben sich Hinweise, daß der Grad an struktureller Ähnlichkeit zwischen Beispiel- und Zielproblemen die Schemainferenz beeinflußt (Pirolli & Anderson in [4]). Diese Annahme haben wir auf die Induktion von Schemahierarchien erweitert und in einem Lernexperiment überprüft. Als Problembereich wurde der Erwerb rekursiver Programmiertechniken verwendet. Die strukturelle Ähnlichkeit zwischen Beispiel- und Zielproblemen wurde systematisch in fünf Stufen variiert. Zusätzlich wurde die Wirkung von Vergleichs- und Anpassungsprozessen getrennt erfaßt, indem einer Gruppe von Probanden das Ergebnis des Vergleichsprozesses vorgegeben wurde.

55 Probanden nahmen an vier aufeinanderfolgenden Tagen jeweils 3 Stunden an der Untersuchung teil. Nach Vermittlung der Syntax einer selbstentwickelten einfachen funktionalen Sprache sollten die Probanden am dritten und vierten Tag sechs rekursive Funktionen am Rechner programmieren. Dabei wurde ihnen je ein Beispiel entsprechend der experimentellen Bedingungen vorgegeben. Geprüft wurde der Einfluß der Lernbedingung auf (1) den Prozeß der Aufgabenlösungen und (2) den resultierenden Wissenserwerb. Der Wissenserwerb wurde dabei zum einen durch einen Abschlußtest erfaßt, bei dem verschiedene Transferaufgaben zu lösen waren. Zum anderen wurde die Schemainduktion explizit erfaßt, indem die Probanden zu drei Zeitpunkten jeweils 28 Funktionen sortieren sollten.

Es zeigte sich, daß die Lernbedingung einen bedeutsamen Einfluß auf die Aufgabenlösung hat: Je ähnlicher ein Beispiel dem Zielproblem ist, desto wahrscheinlicher wird die Aufgabe gelöst; die Vorgabe des Vergleichsprozesses führt ebenfalls zu einer erhöhten Lösungswahrscheinlichkeit. Bezüglich des Wissenserwerbs zeigt die Lernbedingung keinen eindeutigen Einfluß. Betrachtet man die Merkmale nach denen die Probanden sortieren, zeigt sich, daß Probanden, die das Merkmal "Rekursionstyp" zur Klassifikation verwenden, sowohl mehr Aufgaben lösen als auch eine höhere Transferleistung im Abschlußtest erzielen.

Literatur

1. Anderson, J.R. & Thompson, R. (1989). Use of analogy in a production system architecture. In S. Vosniadou and A. Ortony (Eds.), Similarity and Analogical Reasoning, 267-297. Cambridge, Mas.: Cambridge University Press.
2. Carbonell, J.G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Ed.), Machine Learning - An Artificial Intelligence Approach (Vol. 1, pp. 137-162). New York: Springer.
3. Novick, L.R. & Holyoak, K.J. (1991). Mathematical problem solving by analogy. Journal of Experimental Psychology: Learning, Memory, and Cognition, 17, (3) 398-415.
4. Pirolli, P. & Anderson, J.R. (1985). The role of learning from examples in the acquisition of recursive programming skills. Canadian Journal of Psychology, 39, 240-272.
5. Vorberg, D. & Goebel, R. (1991). Das Lösen rekursiver Programmierprobleme: Rekursionsschemata. Kognitionswissenschaft, 1, 83-95.