

# Kopfrechnen aus der Sicht der ACT-R Theorie

Wolfgang Schoppek, Universität Bayreuth

In diesem Kapitel sollen die vielfältigen Befunde aus dem Gebiet des Kopfrechnens einer einheitlichen theoretischen Erklärung zugeführt werden. Die Grundlage dafür bildet die ACT-R Theorie von Anderson & Lebiere (1998). Diese Theorie enthält detaillierte Annahmen über den Erwerb kognitiver Fertigkeiten, was sie besonders geeignet erscheinen lässt, sie auf das Kopfrechnen anzuwenden.

## ACT-R

ACT-R ist eine sogenannte Kognitive Architektur, d.h. sie macht grundlegende Annahmen über kognitive Strukturen und Prozesse, die bei praktisch jeder Denk- und Erkenntnistätigkeit eine Rolle spielen. Kognitive Strukturen sind z.B. das Gedächtnis und die Wahrnehmungssysteme. Kognitive Prozesse betreffen z.B. die Umformung von physikalischen Reizen in Gedächtnisrepräsentationen (Wahrnehmung, Enkodierung), die Speicherung und den Abruf von Informationen. ACT-R liegt als Computerprogramm vor, mit dessen Hilfe Simulationsmodelle menschlichen Denkens erstellt werden können. Eines dieser Modelle spielt für unseren Zusammenhang eine besondere Rolle: die Lebenszeit-Simulation des Erwerbs von Kopfrechenfertigkeiten von Lebiere (1999). Dieses Modell bildet die Entwicklung des Kopfrechnens über viele Jahre hinweg nach (in einem Zeitraffer-Verfahren) und reproduziert damit alle wichtigen empirischen Befunde aus diesem Gebiet.

## Deklaratives und prozedurales Wissen

Eine Hauptannahme von ACT-R besteht in der Unterscheidung zwischen deklarativem und prozeduralem Wissen. Den beiden Wissensarten sind jeweils eigene Gedächtnisstrukturen zugeordnet, die allerdings eng aufeinander bezogen sind. Deklaratives Wissen repräsentiert Fakten; das können relativ abstrakte Bedeutungen sein, aber auch wahrnehmungsnahe Repräsentationen der Wirklichkeit. Fakten im Bereich des Kopfrechnens sind beispielsweise „ $3+4 = 7$ “ oder das Kommutativgesetz „ $a+b = b+a$ “. Die Fakten müssen nicht notwendigerweise wahr sein. So kann z.B. auch das falsche Faktum „ $5+3=7$ “ im Gedächtnis gespeichert sein. Das deklarative Gedächtnis wird von einem Netzwerk hierarchisch verschachtelter Gedächtniselemente, den sogenannten *Chunks*, gebildet. Der Name „Chunk“ wird in der ACT-R Theorie und auch im folgenden Text ganz allgemein zur Bezeichnung deklarativer Gedächtniselemente verwendet. Die Bedeutung des Begriffes weicht also etwas von der spezielleren Bedeutung im Zusammenhang mit der Kapazität des Kurzzeitgedächtnisses ab (zur Erinnerung: Miller hatte die Kapazität des KZG auf  $7 \pm 2$  Chunks geschätzt).

Chunks sind in ACT-R schemaartige Strukturen mit Feldern, die normalerweise andere Chunks enthalten, wodurch die Vernetzung entsteht. Betrachten wir als Beispiel die Repräsentationen des Faktums „ $3+4=7$ “ und der Zahl 3 (die Darstellung entspricht weitgehend der Syntax der ACT-R Software):

```
3+4=7 isa arithmetic-fact
  operand1 three
  operand2 four
  operator plus
  result seven
```

```
three isa number
  image 3
  successor four
```

Die Beispiele zeigen, dass jeder Chunk Vertreter eines bestimmten Typs ist (arithmetic-fact bzw. number), durch den seine Felder festgelegt sind. Die Felder der Chunks enthalten z.T. andere Chunks. Das operand1-Feld des Chunks „3+4=7“ enthält den Chunk „three“, dieser wiederum enthält in seinem successor-Feld den Chunk „four“. Das image-Feld des Chunk-typs number verweist auf das Wahrnehmungsbild der Zahl 3. Alternativ kann das Beispiel auch in Form eines propositionalen Netzwerkes dargestellt werden. Die Abbildung 1 zeigt einen Ausschnitt aus einem solchen Netzwerk. Wie üblich ist die hierarchische Ordnung (isa- oder ist-ein-Relation) vertikal dargestellt und die Nebenordnung (Bestandteil-Relationen wie operand1) horizontal. Auf die Bedeutung der Vernetzung von Chunks werden wir später zurückkommen.

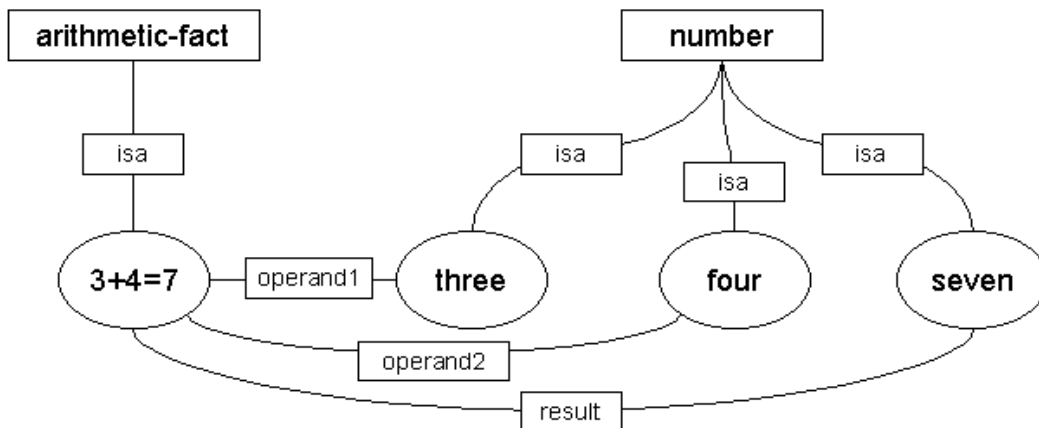


Abbildung1: Ausschnitt aus einem arithmetischen Fakten-Netzwerk in der Darstellungsform eines propositionalen Netzwerkes (Erläuterungen im Text)

*Prozedurales Wissen* ist die Grundlage der kognitiven Prozesse. Es wird üblicherweise als Repräsentation von Vorgängen charakterisiert. Dies führt allerdings leicht zu Missverständnissen, weil es die Vorstellung weckt, dass prozedurales Wissen eine Beschreibung von Vorgängen ist. Es ist aber eher als Programm von Vorgängen anzusehen. Am deutlichsten wird dies, wenn man motorische Fertigkeiten als Analogie heranzieht. Wenn jemand jonglieren kann, wird er beschreiben können, wie man das macht und worauf man besonders achten muss. Das Wissen, das dieser Beschreibung zugrunde liegt, ist deklaratives Wissen – niemand kann aufgrund einer solchen Beschreibung auf Anhieb jonglieren. Prozedurales Wissen dagegen ist das Wissen, das die Ausführung des Jonglierens steuert.

Bausteine des prozeduralen Wissens sind *Produktionsregeln* (oder kurz: *Produktionen*), die Bedingungen mit Aktionen verbinden. Im Bedingungsteil wird auf Inhalte des *Arbeitsgedächtnisses* (AG) Bezug genommen. Wenn die im Bedingungsteil formulierte Konstellation im AG vorliegt, wird der Aktionsteil ausgeführt. Grundsätzlich enthält das AG einen Chunk, der das aktuelle Ziel repräsentiert; zusätzlich Chunks, die das momentan Wahrgenommene und/oder frisch aus dem Gedächtnis abgerufene Information repräsentieren. Der Aktionsteil kann folgende Prozesse ausführen:

- Veränderung des Inhaltes des AG, z.B. Veränderung von Zielen oder Füllen der Felder eines Chunks mit anderen Chunks
- Abruf von Chunks aus dem deklarativen Gedächtnis
- Erzeugen neuer Chunks
- Verschiebung der Aufmerksamkeit
- Vergabe eines motorischen Befehls

Betrachten wir als Beispiel zwei Produktionsregeln, die zu einer gegebenen Additionsaufgabe die Lösung aus dem Gedächtnis abrufen (die Formulierung ist hier natürlich-sprachlich und weicht von der ACT-R Syntax ab<sup>1)</sup>)

Wenn das Ziel ist, eine Additionsaufgabe zu lösen  
und die erste Zahl drei ist  
und die zweite Zahl vier ist (Bedingungsteil)

Dann suche im Gedächtnis nach einem Additions-Faktum  
dessen erste Zahl drei ist und  
dessen zweite Zahl vier ist (Aktionsteil)

Wenn das Ziel ist, eine Additionsaufgabe zu lösen  
und die erste Zahl drei ist  
und die zweite Zahl vier ist  
und im AG ein entsprechendes Additions-Faktum vorliegt (Bedingungsteil)

Dann gib als Antwort das Ergebnis des Additions-Faktums  
und vermerke das Ziel als erreicht (Aktionsteil)

Das Beispiel macht deutlich, wie eng prozedurales und deklaratives Wissen aufeinander bezogen sind, obwohl sie unterschiedlichen Gedächtnisstrukturen zugeordnet werden. Bedingungs- und Aktionsteile von Produktionsregeln enthalten immer Angaben über Zustände des deklarativen Gedächtnisses.

## Symbolische und subsymbolische Ebene

Eine weitere zentrale Annahme in der ACT-R Theorie ist die Unterscheidung zwischen der *symbolischen Ebene* und der *subsymbolischen Ebene* der Verarbeitung. Alles was bislang beschrieben wurde, nämlich der Aufbau und Inhalt von Chunks und Produktionsregeln, ist auf der symbolischen Ebene angesiedelt. Auf dieser Ebene ist alles in Form von ja/nein-Variablen darstellbar: entweder es gibt Chunk A oder es gibt ihn nicht; entweder Produktion B enthält die Bedingung c oder nicht. Die subsymbolische Ebene dagegen beschreibt die Gedächtniselemente mit Hilfe von kontinuierlichen Variablen. So hat jeder Chunk eine bestimmte *Stärke* und eine momentane *Aktivität*. Chunks sind über *Assoziationen* unterschiedlicher Stärke miteinander verbunden. Im deklarativen Gedächtnis haben alle subsymbolischen Parameter mit der Aktivierung von Chunks und deren Ausbreitung im Gedächtnis-Netzwerk zu tun. Den Produktionsregeln sind Parameter der Erfolgswahrscheinlichkeit und der geschätzten Zeitdauer zugeordnet. Im prozeduralen Gedächtnis hat die subsymbolische Ebene also mit dem Erfolg bzw. Misserfolg von Produktionsregeln zu tun und beeinflusst die Auswahl von Regeln im Fall von Konflikten (d.h. wenn bei mehr als einer Produktionsregel der Bedingungsteil erfüllt ist).

Grundlegende Lernprozesse setzen an der subsymbolischen Ebene an. So werden z.B. Chunks mit jedem Gebrauch, also mit jedem Abruf aus dem Gedächtnis, gestärkt, Assoziationen zwischen Chunks verändern sich erfahrungsabhängig, und Produktionen werden durch erfolgreiche Anwendung gestärkt. Diese Lernprozesse laufen automatisch ab, ohne dass eine Lernabsicht vorliegen muss. Für unsere Zwecke genügt es, die Parameter „*Stärke eines Chunks*“ (im Original „baselevel activation“) und „*Assoziationsstärke*“ (zwischen Chunks) zu kennen.

## Stärke eines Chunks

Die Stärke eines Chunks hängt von seiner Nutzungsgeschichte ab: Je öfter ein Chunk benutzt wurde und je kürzer die letzten Nutzungen zurückliegen, desto stärker ist er. Die Stärke eines Chunks beeinflusst, wie leicht er aus dem Gedächtnis abgerufen werden kann (s.u.). Von Nutzung eines Chunks spricht man vor allem, wenn er aus dem Gedächtnis abgerufen wird, aber auch wenn ein Chunk neu erzeugt wird, ist das ein Nutzungs-Ereignis.<sup>ii</sup>

Ein Beispiel aus dem Bereich des Kopfrechnens soll dies verdeutlichen. Nehmen wir an, ein Kind lernt an einem Tag die neuen Fakten „ $4+2=6$ “ und „ $5+3=8$ “ und bildet die entsprechenden Chunks A und B. Wenn nun in den folgenden zwei Wochen „ $4+2=6$ “ zehn mal vorkommt, und „ $5+3=8$ “ nur sechs mal, dann ist Chunk A, der „ $4+2=6$ “ repräsentiert, stärker als Chunk B, der „ $5+3=8$ “ repräsentiert. Oder aber, wenn „ $4+2=6$ “ in zwei Wochen zehn mal vorkommt, „ $5+3=8$ “ dagegen in drei Wochen zehn mal, ist am Ende ebenfalls Chunk A stärker als Chunk B.

## Assoziationsstärke

Zum Verständnis der Assoziationsstärke muss man zunächst wissen, dass sich im Netzwerk des deklarativen Gedächtnisses Aktivierung ausbreitet, die z.B. von den Inhalten des Arbeitsgedächtnisses ausgeht. Grundsätzlich geben Chunks, die eine sogenannte Quellaktivität aufweisen (also eine Quelle von Aktivierung sind), Aktivität an diejenigen Chunks weiter, die die Quellchunks in ihren Feldern enthalten. Die Abbildung 2 veranschaulicht dies. Nehmen wir an, die Chunks „three“ und „four“ sind bereits im Arbeitsgedächtnis, z.B. weil die Aufgabe „ $3 + 4 = ?$ “ gerade gelesen und verstanden wurde. Sie werden damit zu Aktivierungsquellen und geben damit Aktivität an die sie enthaltenden Chunks weiter. In Abbildung 2 sind das „ $3+4=7$ “ und „ $3+5=8$ “. Die Aktivierung breitet sich also gewissermaßen nach oben aus. Das Ausmaß, in dem Aktivierung weitergegeben wird, wird von der Assoziationsstärke bestimmt. Nehmen wir zunächst vereinfachend an, dass die Assoziationsstärken in Abb. 2 alle gleich seien. Dann empfängt der Chunk „ $3+4=7$ “ Aktivität von zwei Quellchunks, der Chunk „ $3+5=8$ “ nur Aktivität von einem Quellchunk und wird damit schwächer aktiviert.

Nach der ACT-R Theorie hängen aber die Assoziationsstärken zwischen zwei Chunks  $j$  und  $i$  davon ab, in wie vielen anderen Chunks Chunk  $j$  enthalten ist. Je häufiger Chunk  $j$  in anderen Chunks enthalten ist (in einem Feld), desto schwächer ist seine Assoziationsstärke zu

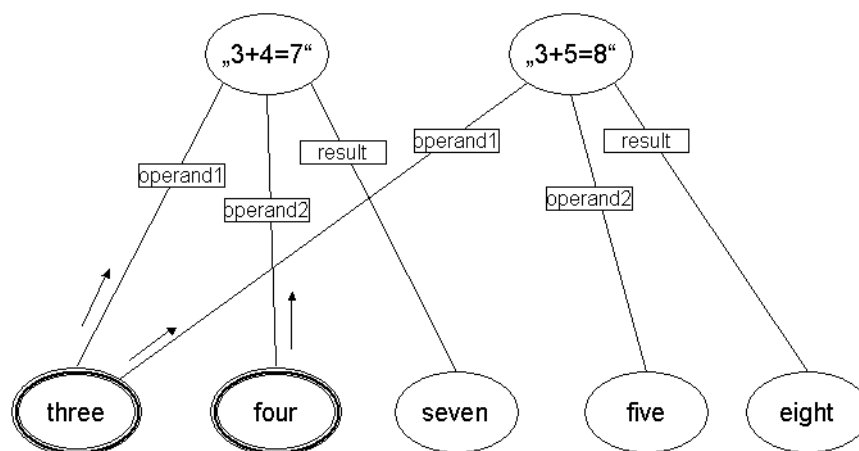


Abbildung 2: Aktivierung der Chunks „ $3+4=7$ “ und „ $3+5=8$ “ durch die Chunks „three“ und „four“, die als Bestandteile des Arbeitsgedächtnisses Quellen von Aktivität sind. Die Aktivität breitet sich in Pfeilrichtung aus.

diesen Chunks. In unserem Beispiel ist Chunk „three“ in zwei anderen Chunks enthalten, Chunk „four“ nur in einem anderen. Dementsprechend ist die Assoziationsstärke zwischen „three“ und „3+4=7“, oder zwischen „three“ und „3+5=8“ niedriger als die zwischen „four“ und „3+4=7“. Wie die Aktivierungsverhältnisse genau aussehen, kann durch entsprechende Gleichungen berechnet werden, die hier nicht genauer betrachtet werden müssen<sup>iii</sup>. Ebenfalls nicht behandelt wird hier der Lernmechanismus, bei dem die Assoziationsstärken sich durch Erfahrung ändern.

### **Aktuelle Aktivität eines Chunks und Abrufwahrscheinlichkeit**

Was letztlich die Wahrscheinlichkeit eines Chunks  $i$  aus dem Gedächtnis bestimmt, ist seine aktuelle Aktivität  $A_i$  (d.h. die Aktivität zum Zeitpunkt des Abrufversuches). Je aktiver der Chunk, desto höher ist die Wahrscheinlichkeit, dass er im Gedächtnis „gefunden“ wird. Die aktuelle Aktivität setzt sich nach Gleichung 1 aus der Stärke  $B_i$  des Chunks und der Netzaktivierung additiv zusammen. Die Netzaktivierung ist diejenige Aktivierung, die der Chunk von seinen Bestandteilen aus dem Netzwerk erhält.

$$(1) \quad A_i = B_i + \sum W_j S_{ji} \quad ,$$

wobei  $W_j$  die Quellaktivität von Chunk  $j$  ist, und  $S_{ji}$  die Assoziationsstärke zwischen Chunk  $j$  und Chunk  $i$ . Der Ausdruck mit dem Summenzeichen bedeutet, dass die Quellaktivität von allen Bestandteilen des Chunks  $i$ , gewichtet durch die jeweilige Assoziationsstärke, aufsummiert wird.

### **Zusammenfassung**

Folgende Sätze fassen die wesentlichen Aussagen der ACT-R Theorie zusammen:

1. Menschliches Wissen wird in deklaratives Wissen und prozedurales Wissen eingeteilt.
2. Deklaratives Wissen repräsentiert Fakten. Auch Meinungen und „falsches Wissen“ wären in diesem Sinne Fakten.
3. Die Elemente des deklarativen Gedächtnisses werden Chunks genannt. (Vorsicht: der Begriff unterscheidet sich etwas vom üblichen „Chunk“-Begriff). Sie enthalten die Bestandteile eines Faktums in Feldern, die normalerweise andere Chunks enthalten.
4. Prozedurales Wissen repräsentiert kognitive Prozesse. Produktionsregeln steuern den Ablauf des Denkens.
5. Produktionsregeln sind die Elemente des prozeduralen Gedächtnisses. Sie bestehen aus einem Bedingungsteil, der Zustände des Arbeitsgedächtnisses beschreibt, bei deren Vorliegen der zugehörige Aktionsteil ausgeführt wird.
6. Die Strukturen von Chunks und von Produktionsregeln werden der symbolischen Ebene zugerechnet. Chunks und Produktionsregeln sind selbst Symbole, die gegenseitig aufeinander Bezug nehmen.
7. Die subsymbolische Ebene beschreibt die Gedächtniselemente und deren Beziehungen mit Hilfe von kontinuierlichen Parametern wie Stärke, Assoziationsstärke, oder Erfolgswahrscheinlichkeit.
8. Lernen findet auf beiden Ebenen statt, auf der symbolischen Ebene indem neue Chunks oder Produktionen gebildet werden, auf der subsymbolischen Ebene, indem der Gebrauch vorhandener Elemente deren Parameter verändert.

9. Die Stärke eines Chunks wächst mit der Häufigkeit seiner Benutzung und nimmt in Zeiten, in denen er nicht genutzt wird ab.
10. Im Netzwerk des deklarativen Wissens breitet sich Aktivität ausgehend von Quellchunks aus. Quellchunks sind im wesentlichen die Elemente des Arbeitsgedächtnisses.
11. Die Aktivität eines Chunks setzt sich zusammen aus seiner Stärke und der Aktivierung, die er über assoziative Verbindungen aus dem Netzwerk empfängt.
12. Je höher die Aktivität eines Chunks ist, desto wahrscheinlicher ist es, dass er aus dem Gedächtnis abgerufen werden kann und desto schneller geht der Abruf.
13. Die Assoziationsstärke zwischen zwei Chunks hängt davon ab, wie oft ein Chunk in den Feldern anderer Chunks enthalten ist. Ist er in vielen anderen Chunks enthalten, nimmt die Assoziationsstärke ab („Fächereffekt“).

## Empirische Befunde zum Kopfrechnen

Die empirischen Befunde zum Kopfrechnen können grob in zwei Klassen eingeteilt werden. Zur ersten Klasse werden Befunde gerechnet, nach denen es zwei grundlegende Strategien<sup>1</sup> zur Lösung von Aufgaben gibt: die Gedächtnisabruf-Strategie und Berechnungs-Strategien. Bei der Gedächtnisabruf-Strategie wird zu einer Aufgabe die richtige Antwort einfach aus dem Gedächtnis abgerufen. Bei Berechnungs-Strategien wird das Ergebnis berechnet, z.B. bei Additionsaufgaben durch „Hochzählen“ unter Zuhilfenahme der Finger, oder bei Multiplikationsaufgaben durch fortgesetzte Addition. Zu dieser Befundklasse gehören auch Ergebnisse und Erklärungen darüber, wovon die Auswahl einer Strategie abhängt.

Die zweite Klasse fasst Befunde zum Effekt der Aufgabengröße zusammen, wonach Aufgaben mit größeren Zahlen längere Zeit zur Lösung benötigen. Bei Verwendung von Berechnungs-Strategien liegt die Erklärung des Effekts auf der Hand. Der Effekt kann aber auch noch bei Erwachsenen nachgewiesen werden, die praktisch keine Berechnungs-Strategien mehr anwenden. Wir werden sehen, dass der Effekt der Aufgabengröße bei Erwachsenen auch über die unterschiedliche Häufigkeit der entsprechenden Aufgaben erklärt werden kann. Im folgenden werden einige wichtige Experimente genauer dargestellt.

Ashcraft (1987) untersuchte die Antwortzeiten von Erwachsenen für Additionsaufgaben, in Abhängigkeit von der Aufgabengröße (Summe), aufgeteilt nach Zwillingaufgaben (zwei gleiche Summanden, z.B.  $6+6=?$ ), Aufgaben mit 0 als Summand, und sonstige (oder normale) Aufgaben. Die Abbildung 3a zeigt die Ergebnisse, nach denen bei normalen Aufgaben

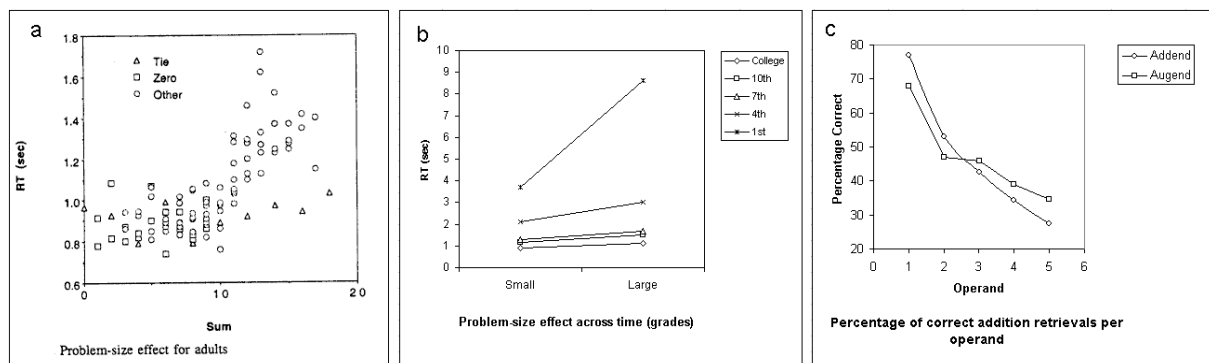


Abbildung 3: Empirische Befunde zum Effekt der Aufgabengröße

<sup>1</sup> Der Begriff Strategie bezeichnet in diesem Zusammenhang einfach eine Vorgehensweise.

die Antwortzeiten mit wachsender Summe ansteigen, dies auch bei Zwillingaufgaben zu beobachten ist, allerdings mit flacherem Anstieg, und die Antwortzeiten bei Null-Aufgaben nicht von der Aufgabengröße abhängen. In einem anderen Experiment erfasste Ashcraft (1987) den Effekt der Aufgabengröße nach Jahrgangsstufen getrennt und stellte fest, dass der Effekt mit zunehmendem Alter schwächer wird (Abbildung 3b).

Siegler & Shrager (1984) ließen 4-jährige Kinder Aufgaben zwischen 1+1 und 5+5 ausrechnen und protokollierten alle Antworten. Im Anteil richtiger Lösungen ergab sich ein deutlicher Effekt der Aufgabengröße, der für den zweiten Summanden ausgeprägter ist als für den ersten Summanden (Abbildung 3c). Ähnlich sehen auch die Ergebnisse eines Experimentes aus, bei dem Viertklässler Multiplikationsaufgaben zu lösen hatten (Siegler, 1988).

## Das Modell zur Entwicklung des Kopfrechnens von Lebiere (1999)

Das ACT-R Modell von Lebiere (1999) besteht zu Beginn aus wenigen Produktionsregeln und deklarativen Fakten. Es simuliert den Erwerb von Wissen durch die Bearbeitung von Aufgaben zunehmender Schwierigkeit über viele Jahre hinweg. Dabei werden Zeiten, in denen keine Aufgaben bearbeitet werden (also ein Großteil der Lebenszeit) einfach rechnerisch berücksichtigt, wodurch diese Simulation erst möglich wird.

Im deklarativen Gedächtnis des Modells sind anfangs nur die Zahlen und deren Abfolge gespeichert, also zu jeder Zahl ihr Nachfolger. Das Modell verfügt auch über einige Produktionsregeln: Erstens kann das Modell Aufgaben „verstehen“, indem es die Gegebenheiten

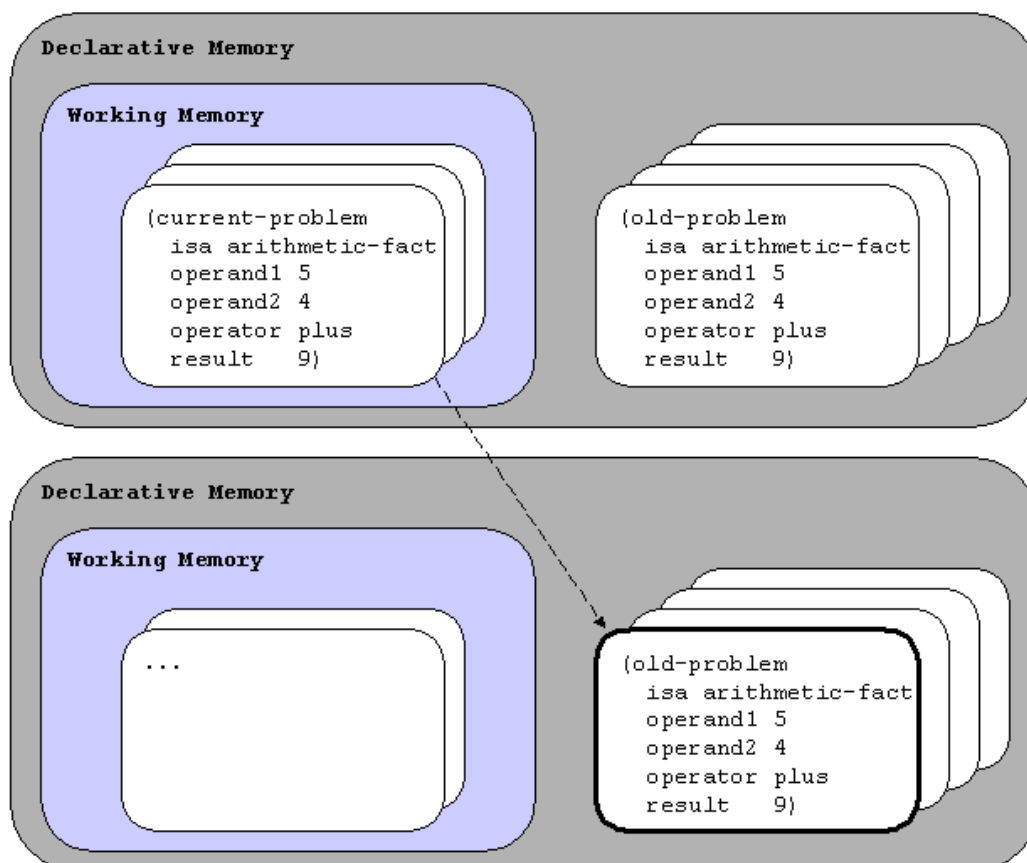


Abbildung 4: Die Abbildung veranschaulicht, wie ein Chunk, der eine gelöste Aufgabe repräsentiert (current-problem) am Ende mit einem bereits im Gedächtnis gespeicherten, identischen Chunk (old-problem) verschmolzen wird, wobei dieser gestärkt wird (angedeutet durch die dickere Linie). Dargestellt ist auch, dass das Arbeitsgedächtnis (Working Memory) ein Teil des deklarativen Gedächtnisses ist (Declarative Memory).

einer Aufgabe richtig in entsprechende Chunks übersetzt, bei denen allerdings das Ergebnisfeld leer ist. Zweitens verfügt das Modell über iterative Berechnungs-Strategien, die sich die Kenntnis der Abfolge der Zahlen zunutze machen. Im Fall der Addition wird der Höhe des zweiten Summanden entsprechend hochgezählt. Drittens kann das Modell bereits im Gedächtnis nach passenden „arithmetischen Fakten“ suchen (auch wenn es zu Beginn dort nichts finden wird). Eine eigene Produktionsregel ist für die Lösung von Null-Problemen zuständig. Sie bestimmt als Ergebnis einfach den Summanden, der ungleich Null ist. Eine weitere Produktion gibt die Antwort aus und „entlässt“ die bearbeitete Aufgabe aus dem Arbeitsgedächtnis.

Da die Aufgaben in Chunks desselben Typs kodiert werden wie die arithmetischen Fakten, bildet jede gelöste Aufgabe ein neues Faktum, das prinzipiell wieder aus dem deklarativen Gedächtnis abgerufen werden kann. Im Fall von wiederholten Aufgabenlösungen werden nicht etwa identische Chunks im Gedächtnis angehäuft, sondern der neue Chunk mit dem bereits vorhandenen identischen Chunk verschmolzen, wobei der vorhandene Chunk so gestärkt wird, als ob er aus dem Gedächtnis abgerufen worden wäre (siehe Abbildung 4). Nach den ersten paar Bearbeitungen derselben Aufgabe – zu Beginn natürlich mit der Berechnungs-Strategie – sind die resultierenden arithmetischen Fakten noch so schwach im Gedächtnis verankert, dass ihre Abrufwahrscheinlichkeit sehr gering ist. Mit zunehmenden Wiederholungen werden die entsprechenden Chunks aber so gestärkt, dass sie aus dem Gedächtnis abgerufen werden können und somit für die Gedächtnisabruf-Strategie zur Verfügung stehen. Das Modell simuliert also die Entwicklung von Rechenfertigkeiten dadurch, dass zunächst alle Aufgaben mit der Berechnungs-Strategie bearbeitet werden. Dadurch sammeln sich Chunks im deklarativen Gedächtnis an, die arithmetische Fakten repräsentieren. Mit zunehmender Übung werden mehr und mehr dieser Chunks so gestärkt, dass Sie direkt abgerufen werden können und damit die Gedächtnisabruf-Strategie ermöglichen.

Wichtig für die Vorhersagen des Modells ist, dass vor allem die Prozesse der Aktivierung Zufallsschwankungen unterliegen. Das führt manchmal dazu, dass von zwei etwa gleich starken Chunks, die beispielsweise zu einer Zahl den richtigen bzw. den falschen Nachfolger enthalten, auch einmal der „falsche“ abgerufen wird. Das führt z.B. beim Anwenden der Berechnungs-Strategie dazu, dass die Kinder zu einem falschen Ergebnis kommen. Nach den Annahmen der ACT-R Theorie wird auch diese falsche Lösung im deklarativen Gedächtnis gespeichert und ist potentiell abrufbar. Auch wenn die Chunks mit falschen Lösungen meist schwächer sind als die mit den richtigen, ist deren Vorhandensein eine Fehlerquelle. Auch dabei spielen wieder die Zufallsschwankungen eine Rolle (s.o.). Als Konsequenz aus der Annahme dass auch falsche Ergebnisse gespeichert werden ergibt sich, dass vor allem in frühen Phasen des Lernens darauf geachtet werden muss, dass Rechenfehler möglichst sofort korrigiert werden.

## **Erklärung der Befunde durch das Modell von Lebiere (1999)**

Wie können nun die zuvor berichteten Befunde mit dem Modell von Lebiere (1999) erklärt werden? Ein Simulationsmodell erzeugt Verhaltensdaten, die mit den gleichen Methoden ausgewertet werden wie die Daten von Versuchspersonen, womit die Daten direkt verglichen werden können. Bei hoher Übereinstimmung wird das Modell als Erklärung für die Befunde angesehen. Darüber hinaus ist es wichtig zu verstehen, aufgrund welcher Bestandteile und Prozesse das Modell die Daten vorhersagt bzw. nachbildet. In Lebiere's „Lebenszeitsimulation“ bearbeitete das Modell pro simuliertem Jahr 4000 Rechenaufgaben, die sich an Schulbuchanalysen von Ashcraft (1987) und Ashcraft und Christy (1995) anlehnen.

Die Strategievarianten „Gedächtnisabruf“ und „Berechnung“ sind direkt in das Modell aufgenommen worden. Der Einfachheit halber nimmt Lebiere an, dass immer zuerst die Gedächtnisstrategie versucht wird, und erst wenn kein passender Chunk abgerufen werden kann zur Berechnungs-Strategie übergegangen wird. Auch wenn diese Annahme zumindest für Anfänger unrealistisch zu sein scheint, beeinträchtigt sie die guten Vorhersagen des Modells nicht.



Effekte der Aufgabengröße lassen sich bei Anfängern leicht auf die Verwendung der Berechnungs-Strategie zurückführen: Das Hochzählen dauert umso länger, je höher die zu addierende Zahl ist. Dasselbe gilt für die wiederholte Addition einer Zahl beim Multiplizieren. Das Schwächerwerden des Effektes mit zunehmender Übung (Jahrgangsstufe) erklärt sich hauptsächlich durch die immer häufigere Verwendung der Abruf-Strategie. Bei dieser Strategie spielt es kaum eine Rolle, wie hoch die zu addierenden Zahlen sind. Wenn zwei Chunks gleich stark sind, ist es für die Zeit, die der Abruf benötigt gleichgültig, ob die Zahlen 4, 3, und 7 sind oder 7, 6, und 13 – es wird in beiden Fällen nur ein Chunk abgerufen.

Es bleibt aber immer noch die Frage, wie man den minimalen Effekt der Aufgabengröße erklären kann, der selbst bei Erwachsenen noch zu beobachten ist. Im Modell von Lebiere genügt zur Erklärung die Tatsache, dass Aufgaben umso seltener vorkommen, je größer die beteiligten Zahlen sind (dies wurde empirisch festgestellt). Da die entsprechenden Chunks seltener gebraucht werden sind sie auch schwächer (siehe Satz 9 der Zusammenfassung). Damit dauert auch ihr Abruf aus dem Gedächtnis länger (Satz 12).

Das Ergebnis von Ashcraft (1987), nach dem Additions-Aufgaben mit Null als Summand etwa gleich schnell gelöst werden, egal wie hoch der andere Summand ist erklärt das Modell mit dem Vorhandensein einer eigenen Produktionsregel für diese Aufgabenart. Da diese Regel als Ergebnis einfach den anderen Summanden einsetzt, beinhaltet sie nicht einmal einen Abruf aus dem deklarativen Gedächtnis. Komplizierter zu erklären ist das Phänomen, dass auch Zwillingsaufgaben schneller bearbeitet werden als „normale“ Aufgaben. Lebiere greift dabei zu einem Trick, indem er annimmt, dass solche Aufgaben besonders kodiert werden und zwar in Form von „x doppelt“ anstatt „x+x“. Da der Chunk „doppelt“ nur in vergleichsweise wenigen arithmetischen Fakten vorkommt, ist seine Assoziationsstärke zu diesen Fakten auch entsprechend größer. Wenn also z.B. die Aufgabe  $6+6=?$  gestellt wird, wird diese als „6 doppelt“ kodiert, und die Chunks „6“ und „doppelt“ werden zu Quellen von Aktivität. Durch die vergleichsweise hohe Assoziationsstärke des Chunks „doppelt“ wird das Faktum „6 doppelt = 12“ besonders stark aktiviert, was zu einem schnelleren Abruf führt. Wenn auch diese Erklärung nicht besonders überzeugend ist, gibt es immerhin empirische Belege dafür, dass Menschen Zwillingsaufgaben in einer besonderen Weise verarbeiten, da sie für die Kodierung der Aufgaben länger brauchen (Eliaser, Siegler, Campbell, & Lemaire, in preparation).

## **Erklärungsansätze der ACT-R Theorie für Rechenschwierigkeiten**

Obwohl das Modell von Lebiere (1999) keine direkten Aussagen über Rechenschwierigkeiten macht, lassen sich doch viele Befunde aus diesem Bereich damit erklären (zusammenfassend zu diesen Befunden siehe Zielinski, 1995). So demonstriert das Modell deutlich die Abhängigkeit der Rechenfertigkeiten voneinander: Wenn die Reihenfolge der natürlichen Zahlen noch nicht sicher beherrscht wird, führt die Berechnungs-Strategie häufig zu falschen Ergebnissen, was wiederum die erfolgreiche Anwendung der Gedächtnis-Strategie beeinträchtigt. Auch wenn dies bisher nicht angesprochen wurde, enthält das Modell auch die Möglichkeit, dass Strategien falsch gelernt bzw. noch nicht genügend eingeübt wurden (dies betrifft dann das prozedurale Gedächtnis, bzw. die Übertragung von deklarativen Vorgangsbeschreibungen in Produktionsregeln durch Übung). Schwache Rechenleistungen können auch damit erklärt werden, dass zu viele falsche Fakten im deklarativen Gedächtnis gespeichert sind, bzw. dass deren Stärke zu hoch ist.

Die Einteilung in Berechnungs- und Gedächtnisabruf-Strategie stellt natürlich eine Vereinfachung dar, weil es innerhalb jeder Strategiekategorie vielfältige Varianten gibt. Ein wichtiger Entwicklungsschritt in der Verfeinerung der Berechnungs-Strategie besteht z.B. darin, dass die Kinder die Anordnung der Summanden so umstellen, dass sie immer vom größeren aus hochzählen. Auch dazu bietet die ACT-R Theorie Erklärungen an, die hier nicht im Detail besprochen werden können. Es sei nur soviel gesagt, dass sie die Auswahl konkurrierender Strategien in Abhängigkeit von deren jeweiliger „Erfolgsgeschichte“ vorhersagt. Neue Strate-

gien haben es demnach grundsätzlich schwer, sich gegenüber bewährten alten Strategien durchzusetzen (z.B. Abruf gegenüber Berechnung).

Rechnen in der Schule ist natürlich nicht nur Kopfrechnen, aber auch schriftliche Verfahren beinhalten Kopfrechnen. Wer beim Erlernen schriftlicher Verfahren noch nicht sicher Kopfrechnen kann, muss einen großen Teil seiner Aufmerksamkeit dem Berechnen von elementaren Teilschritten widmen, wodurch das Lernen des Verfahrens selbst beeinträchtigt wird.

## Literatur

- Anderson J. R. & Lebiere C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.
- Ashcraft M. H. (1987). Children`s knowledge of simple arithmetic: A developmental model and simulation. In J. Bisanz, C. J. Brainerd, & R. Kail (Eds.), *Formal methods in developmental psychology: Progress in cognitive development research* (pp. 302-338). New York: Springer.
- Ashcraft M. H. & Christy K. S. (1995). The frequency of arithmetic facts in elementary texts: Addition and multiplication in grades 1-6. *Journal for Research in Mathematics Education*, **26**, 396-421.
- Lebiere C. (1999). The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft*, **8**, 5-19.
- Siegler R. S. & Shrager J. (1984). A model of strategy choice. In C. Sophian (Eds.), *Origins of cognitive skills* (pp. 229-293). Hillsdale, NJ: Erlbaum.
- Zielinski W. (1995). *Lernschwierigkeiten*. 2. Auflage. Stuttgart: Kohlhammer.

---

## Endnoten

<sup>i</sup> Für diejenigen, die sich für die genaue ACT-R Syntax interessieren hier eine Original-Version der beiden Produktionsregeln. Die Zahlen drei und vier werden hier als Variablen =x und =y kodiert, so dass diese Produktionsregel für alle möglichen Kombinationen von Zahlen verwendet werden kann.

```
(p retrieve-addition-fact
  =goal>
  isa addition-fact
  operand1 =x
  operand2 =y
  result nil
==>
+retrieval>
  operand1 =x
  operand2 =y
)

(p extract-result
  =goal>
  isa addition-fact
  operand1 =x
  operand2 =y
  result nil
=retrieval>
  isa addition-fact
  operand1 =x
  operand2 =y
  result =z
==>
=goal>
  result =z
+motor>
  isa speak-sentence
  content "Die Antwort ist =z"
!pop!
)
```

<sup>ii</sup> Die vereinfachte Formel zur Berechnung der Stärke ( $B$ ) eines Chunks lautet:

$$B = \ln\left(\frac{2n}{\sqrt{L}}\right),$$

wobei  $n$  die Zahl der Nutzungsereignisse des Chunks ist, und  $L$  das „Alter“ des Chunks in Sekunden.

<sup>iii</sup> Die Formel zur Berechnung der Assoziationsstärke zwischen Chunk  $j$  und Chunk  $i$  ( $S_{ji}$ ) lautet

$$S_{ji} = S - \ln(n),$$

wobei  $n$  die Zahl der Chunks ist, in denen Chunk  $j$  enthalten ist, und  $S$  eine Konstante (ca. 300).