

Eine Einführung in Kognitive Architekturen

Universität des Saarlandes

AI Tools Seminar Wintersemester 06/07

(Dr. Michael Kipp, Dr. Alassane Ndiaye, Dr. Dominik Heckmann, Michael Feld)

Christian Michalk
Quellenstraße 12
66679 Losheim
(C.Michalk@gmx.de)

Abstract

Diese Ausarbeitung vermittelt dem Leser Grundlagenwissen über die Struktur und Arbeitsweise kognitiver Architekturen. Nachdem zunächst die wesentlichen Annahmen über Wissensrepräsentation und Wissensverarbeitung erläutert werden, wird mit dem Produktionssystem ein wesentlicher Bestandteil aller Architekturen eingeführt. Des Weiteren bekommt der Leser einen Überblick über die Arbeitsweise der kognitiven Architekturen. Dabei werden alle wesentlichen Aspekte wie Operatoren und deren Suchräume, Chunking, Impasses und Konflikte, statistisches Lernen, Wissensrepräsentation erläutert. Ein weiteres Kapitel stellt die bekanntesten Architekturen SOAR und ACT-R vor und hebt deren Besonderheiten hervor. Kleine Beispielprogramme runden hier die Kapitel ab. Das letzte Kapitel widmet sich Allen Newells Idee der vereinheitlichten Theorie.

Inhaltsverzeichnis

1.	Einleitung	1
2.	Motivation und Ziele	1
3.	Grundlagen.....	3
3.1	Annahmen und Vereinfachungen.....	4
3.2	Produktionssysteme	4
3.3	Probleme und Suchräume.....	5
4.	Arbeitsweise	6
4.1	Zyklus	6
4.2	Regeln, Regelauswahl und Konfliktlösung	6
4.3	Sackgassen: Impasses	6
4.4	Lernen: Chunking.....	7
5.	Bekannte Architekturen.....	7
5.1	SOAR.....	7
5.2	ACT-R.....	8
6.	Unified Theories of Cognition	10
7.	Zusammenfassung.....	11
8.	Literatur	12

1. Einleitung

Seit dem ersten Aufkommen des Begriffes der Künstlichen Intelligenz strebt die Wissenschaft gezielt danach, die Grundfesten der menschlichen kognitiven Fähigkeiten zu verstehen, um so die Grundsteine für die Erschaffung des vollen Spektrums der menschlichen Intelligenz außerhalb des menschlichen Körpers, vornehmlich im Computer, zu schaffen.

Wenn auch die ersten Verheißungen von Wissenschaftlern euphorisch von der Erschaffung denkender Computersysteme in naher Zukunft sprachen, welche wissenschaftliche Erkenntnisse produzieren und beweisen sollten, so folgte nach den ersten Forschungen eine Stagnation der Euphorie und die Erkenntnis, dass in dieser wissenschaftlichen Disziplin zum Erreichen der Ziele noch ein langer Weg bevorsteht.

Als Produkt der Grundlagenforschung entstanden viele Theorien und Systeme, welche Teilbereiche der menschlichen Intelligenz nachbilden können – hauptsächlich das Lösen von logischen Problemen. Diese Theorien und Systeme unterliegen nicht mehr dem Ziel eine allgemeine, Künstliche Intelligenz zu erzeugen, sondern finden praktischen Einsatz im Alltag bei ganz bestimmten Problemen.

Ein Zweig, der sich noch mit dem ursprünglichen Ziel beschäftigt, die gesamten kognitiven Fähigkeiten des Menschen zu verstehen und zu reproduzieren, ist die Kognitionswissenschaft. Diese versucht Phänomene wie Gedächtnis, Sprache, Wahrnehmung, Problemlösung, usw. als berechenbare Modelle zu formulieren. Diese Modelle werden dann als kognitive Architekturen auf dem Computer realisiert.

Im Folgenden sollen nun die wesentlichen Ziele dieser Forschung, die theoretischen Grundlagen der Modelle und die Arbeitsweise der kognitiven Architekturen erläutert werden. Das zweite Kapitel geht also erst einmal näher auf die Ziele dieser Forschungstätigkeit ein. Danach folgt ein Grundlagenkapitel, welches wesentliche Begriffe und Theorien erklärt. Im vierten Kapitel wird dann die Arbeitsweise der Architekturen behandelt. Das fünfte Kapitel stellt die zwei bekanntesten Architekturen kurz vor. Das sechste Kapitel geht näher auf das Endziel dieser Forschung ein, der so genannten Unified Theorie.

2. Motivation und Ziele

Der Begriff Kognition leitet sich aus dem lateinischen Verb Cognoscere (lat. Erkennen) ab. Architektur ist eine Übersetzung der griechischen Wörtern Αρχη (erstes) und τεχνη (Handwerk). Der Begriff bezeichnet also ein Modell, einen „Bauplan“ des Menschen, in dem genau das enthalten ist, was bei allen denkenden

Individuen gleich und fest ist. Übertragen auf das Modell sind dabei im Wesentlichen die Annahmen über Wissensrepräsentation, Eigenschaften des Speichers, und der Prozess, der auf dem Speicher arbeitet, gemeint. Das Wissen selbst wird dabei nicht als Teil der Architektur angesehen – wie bei einem Computer verschiedene Programme ausgeführt werden können, so können kognitive Architekturen verschiedene Wissensbasen interpretieren.

Im Gegensatz zu spezialisierten Systemen, welche intelligentes Verhalten zum Lösen bestimmter Problemstellungen produzieren, versucht man eine allgemeine Intelligenz auf der Systemebene zu schaffen. Auch wenn nur das Erkennen und Treffen von Entscheidungen für die grundlegende Arbeitsweise kognitiver Architekturen wichtig ist, so braucht man dennoch die gesamte Palette an Fähigkeiten, um das intelligente Verhalten des Menschen vollständig abzudecken.

Ein wesentliches Problem bei Entwurf kognitiver Architekturen ist die Frage, wie möglichst viele Wissensquellen erschlossen werden können, und so gewonnene Erkenntnisse in den Entscheidungsprozess einfließen können. Die meisten kognitiven Fähigkeiten des Menschen bieten ihm den Zugang zu ganz bestimmten Informationsquellen:

- Wahrnehmung und Aufmerksamkeit: Der Mensch bedient sich einer Vielzahl an Informationsquellen und -medien: Wir empfangen Licht über unser Auge und bilden uns daraus eine räumliche Vorstellung unserer Umgebung, wir fühlen Temperaturen und Materialien, wir schmecken und riechen, hören Geräusche und kommunizieren. Simple Temperatursensoren z.B. könnten hier als digitales Gegenstück kontinuierlich Werte übermitteln. Aber auch komplexere Mechanismen wären denkbar – wie stereoskopische Kameras oder Sonarsensoren, mit denen ein Abbild der Umgebung mit Tiefeninformation erzeugt werden kann. Bei der Wahrnehmung spielt aber auch die Informationsverarbeitung eine wichtige Rolle – und somit auch die Zuteilung begrenzter Ressourcen auf die für die Informationsgewinnung wesentlichen Prozesse.
- Gedächtnis und Metawissen: Auch wenn der Informationsstrom über gewisse Dinge längst erloschen ist, so sind wir später in der Lage, viele wichtige Informationen wieder abzurufen und damit unser weiteres Denken und Verhalten zu beeinflussen. Weiterhin speichern wir Wissen über Wissen – z.B. die Zuverlässigkeit oder wie lange Informationen gültig bleiben. Übertragen auf kognitive Architekturen wären dies also die Fähigkeiten die Ergebnisse kognitiver Prozesse zu speichern und diese später wieder an geeigneter Stelle abrufen zu können. Dadurch können kognitive Prozesse durch gespeichertes Wissen beeinflusst werden. Durch Metawissen kann der Einfluss durch bestimmte Informationen wiederum beeinflusst werden. Anhand Erwartungen über bekannte Sachverhalte, kann der kognitive Prozess gezielt in optimalere

Richtungen gesteuert werden. Ein wichtiger Aspekt ist ebenfalls die Fähigkeit zu erkennen, wann bestimmte Informationen fehlen um ein Problem angemessen lösen zu können.

- Kommunikation und Sprache: Nicht nur die gewaltige Ausdruckskraft und die Vielseitigkeit der menschlichen Sprache, sondern auch die nonverbale Kommunikation erlaubt es uns mit anderen Informationen auszutauschen – dies ist oft der schnellste Weg um an gewünschte Informationen zu kommen. Beim Entwurf kognitiver Architekturen muss also auch berücksichtigt werden, dass sich diese nicht alleine in ihrer Umwelt befinden, sondern dass es hier noch andere intelligente Agenten gibt, mit denen durch Kommunikation ein reger Informationsaustausch möglich ist.
- Reflexion und Problemlösen: Sehen wir uns einer Problemstellung gegenüber, so sind wir in der Lage hierzu alle gespeicherten und relevanten Informationen wieder abzurufen und können so aus bereits gespeichertem Wissen erneut lernen. Durch die Analyse vergangener kognitiver Prozesse können Informationen gewonnen werden, um so zukünftige Prozesse positiv zu beeinflussen. Außerdem können Gründe für bestimmte Entscheidungen hergeleitet werden und auf ähnliche Problemstellungen übertragen werden. Durch das Vorhersehen der Auswirkungen möglicher Aktionen auf die Umwelt können Ablaufpläne erstellt werden, die bestimmten Zielen des Agenten gerecht werden.
- Lernen: Lernen beinhaltet u.A. die Generalisierung bestimmter Konzepte über spezifisches Wissen und bestimmte Beobachtungen hinaus – also das Anwenden, Erkennen und Speichern von Lösungen und Verfahren zu bestimmten Situationen. Auch wenn viele Architekturen das Lernen als automatisierten und unbewussten Prozess behandeln, so ist es aber auch möglich durch Schlussfolgerungen über Prozesse in einer bewussteren Art und Weise zu lernen.

3. Grundlagen

Wie bereits in der Einleitung angesprochen, sind die zwei Hauptergebnisse der Forschungsarbeiten Modelle kognitiver Architekturen und deren Realisierung in Form von Computerprogrammen. Zunächst soll hier die Vereinfachung bestimmter Phänomene im Modell besprochen werden. Die Verwendung von Computer verlangt außerdem eine klare, mathematische Definition aller Arbeitsschritte, die durchgeführt werden sollen. Im Weiteren sollen für kognitive Architekturen typische Verfahren definiert und erklärt werden.

3.1 Annahmen und Vereinfachungen

Jede Form von Intelligenz braucht einen dauerhaften Informationsspeicher bzw. Gedächtnis. Beim Menschen beinhaltet das Gedächtnis alles ruhende Wissen. Dies wären unter Anderem Faktenwissen, erlernte Vorgänge, Erfahrungen, Emotionen und Eindrücke. Im Modell wird dem Gedächtnis ebenfalls Rechnung getragen – allerdings vereinfacht. Es wird strikt unterschieden zwischen deklarativem und prozeduralem Wissen.

Deklaratives Wissen enthält Informationen über Fakten, von denen geglaubt wird, dass sie eine wahre Aussage über gewisse Dinge treffen. Im prozeduralen Wissen werden Ableitungsregeln definiert, mit deren Hilfe aus bereits bestehenden Fakten als Voraussetzung neue Fakten gewonnen werden können. Diese Fakten können dann als Grundlage für weitere Regelanwendungen dienen.

Weiterhin steckt alleine in dem Versuch, intelligente Fähigkeiten des Menschen auf dem Computer durch solche Regelsysteme nachzubilden, die Annahme, dass die menschlichen Fähigkeiten durch entsprechendes Regelwerk reproduzierbar sind. Man also versucht Gedankengänge in Form von symbolischen Wissen und entsprechenden Ableitungen nachzubilden.

3.2 Produktionssysteme

Unter der Annahme, dass Intelligenz etwas Berechenbares ist, reduziert sich die Aufgabe Künstliche Intelligenz auf dem Computer nachzubilden darauf, die richtigen Datenstrukturen und Algorithmen zu finden, die eben genau das gewünschte Verhalten erzeugen. Ein solcher grundlegender Bestandteil der kognitiven Architekturen bilden Produktionssysteme.

Ein Produktionssystem besteht aus einer deklarativen Wissensbasis, den Fakten, und einem prozeduralen Speicher, den Regeln. Die Fakten werden dabei in Form von atomaren Wissenseinheiten gespeichert. Eine Regel besteht aus zwei Teilen: Den Voraussetzungen und dem Effekt. Beide bestehen aus einer Zusammensetzung von Fakten. Die Regeln, deren Voraussetzung durch die Faktenbasis erfüllt werden, bilden die Konfliktmenge. Das System muss aus dieser Menge eine Regel auswählen, deren Effekt als Faktum in den deklarativen Speicher übernommen wird. Dieser Vorgang wird solange wiederholt, bis ein Zielzustand erreicht ist, oder keine weiteren Regeln mehr anwendbar sind. Die Arbeitsweise dabei ist episodisch – es wiederholt sich ständig der Ablauf: Konfliktmenge bilden, Regel wählen, ausführen.

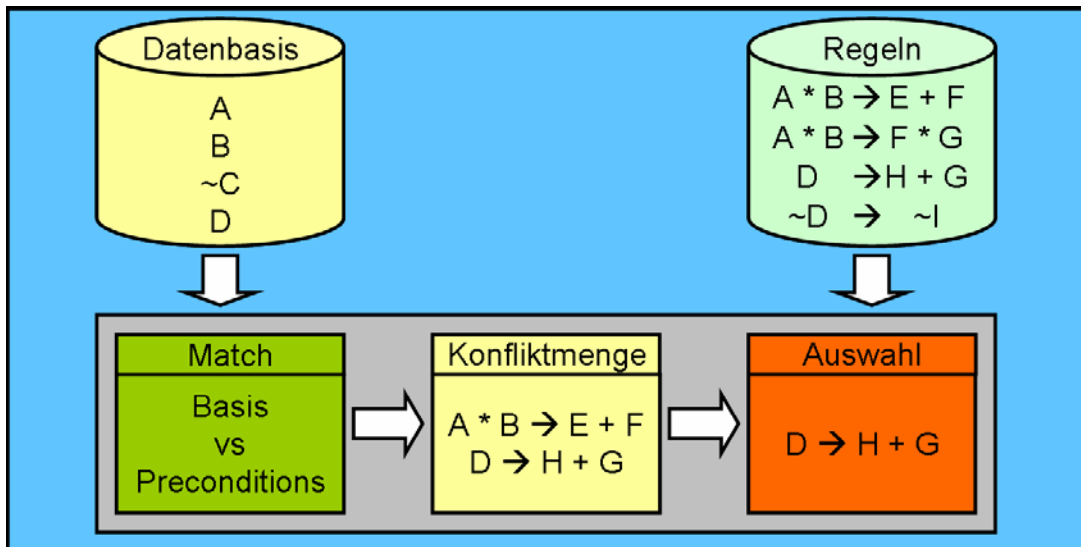


Abbildung 1: Beispieldarstellung einer Wissensbasis, der dazugehörigen Regeln und der zyklischen Vorgehensweise von SOAR.

Die Hauptaufgabe eines Produktionensystems besteht also darin, aus dem Anfangszustand und gegebenen Regeln einen Zustand zu finden, in dem bestimmte Fakten gegeben sind. Dabei unterscheidet man grundlegend zwei Vorgehensweisen – die Suche von der anfänglichen Faktenbasis in Richtung Zielzustand (forward chaining), und die Suche rückwärts von dem gegebenen Zielzustand zurück zur Faktenbasis (backward chaining).

3.3 Probleme und Suchräume

Eine wesentliche Aussage des vorherigen Kapitel besteht also darin, dass man versucht, Problemstellungen als Fakten und Regeln in einem Produktionssystem zu repräsentieren, um so nach einer gewissen Zahl durchlaufener Episoden die Lösung hierfür zu erhalten.

Denkt man aber bei einem gegebenen Problem darüber nach, welche erdrückende Informationsflut einem Menschen dazu einfallen kann, die nichts mit der Lösung des Problems zu tun hat, so wird schnell klar, dass beim Menschen hier eine Beschränkung auf das Wesentliche beim Lösen solcher Probleme stattfindet.

In Analogie hierzu kann man sich leicht vorstellen, dass bei einem gegebenen Problem im Modell durch eine breite Faktenbasis und einem entsprechend umfangreichen Regelsystem ein Suchraum erzeugt wird, welcher die Möglichkeiten moderner Computersysteme weit überschreitet. Die mächtigste Waffe zur Komplexitätsbewältigung besteht in der geschickten Auswahl der Regeln, also der gezielten Steuerung der Suche in Richtung des Zielzustandes.

4. Arbeitsweise

Nachdem nun die wichtigsten Grundlagen behandelt wurden, wird im Folgenden die Arbeitsweise kognitiver Architekturen erläutert.

4.1 Zyklus

Wie bereits erklärt arbeitet im Kern jeder kognitiver Architektur ein Produktionssystem. Die dadurch gegebene episodische Verarbeitung überträgt sich auf die gesamte kognitive Architektur.

Da die Systeme zusätzlich noch Möglichkeiten zur Ein- und Ausgabe bieten, läuft der Zyklus der Systeme in der Reihenfolge *Eingabe, Konfliktmenge bilden, Konfliktlösung, Anwendung, Ausgabe* ab.

4.2 Regeln, Regelauswahl und Konfliktlösung

Regeln bestehen also aus Voraussetzungen und Effekten. Bei kognitiven Architekturen sind die Effekte aber nicht auf das Hinzufügen und Entfernen von Fakten beschränkt – Effekte können auch Aktoren des Agenten steuern und Voraussetzungen können abgefragte Werte von Sensoren sein.

Regeln entsprechen den Handlungsmöglichkeiten, denen sich Menschen in ihrer Umwelt bewusst sind. Dabei spielt die Fähigkeit, relevante von irrelevanten Handlungsmöglichkeiten zu unterscheiden, beim Menschen eine große Rolle. Die entsprechende Herausforderung bei kognitiven Architekturen besteht darin, den komplexen Suchraum zu bewältigen. Hierzu gibt zwei Mechanismen die Regelauswahl bei der Konfliktlösung zu beeinflussen: Metaregeln und statistische Bewertung.

Metaregeln sind Regeln, welche eine Rangreihenfolge unter Operatoren definieren. Stehen nun mehrere Regeln in einem Zyklus zur Auswahl, so kann das System die Regel auswählen, die durch Metaregeln entsprechend bevorzugt wird.

Bei statistischen Verfahren werden über längere Zeit während der Ausführung Regeln und Fakten bewertet. So werden z.B. bei ACT-R für alle Regeln vorher Aktivierungskosten berechnet. Diese Kosten setzen sich zusammen aus erwartetem Nutzen, vergangener Nutzung und Rauschen. Der erwartete Nutzen ergibt sich aus dem Nutzen vergangener, ähnlicher Situationen. Kosten durch zu häufiges Feuern verhindern, dass der Suchprozess monoton in eine Richtung läuft. Mit dem Rauschen wird gezielt zufälligeres Verhalten erzeugt. Die Kosten für vergangene Nutzungen und erwartetem Nutzen werden dabei ständig aktualisiert.

4.3 Sackgassen: Impasses

Trotz aller Mechanismen kann es in einem Zyklus vorkommen, dass mehrere Regeln der Konfliktmenge anwendbar sind, der Architektur aber keine Präferenzen

vorliegen. Um in solchen Situation den bestmöglichen Lösungsweg zu finden, verwenden die Architekturen verschiedene Mechanismen.

SOAR z.B. generiert in diesem Falle einen Substate mit eigenem Subgoal, d.h. beim Aufkommen einer solchen Situation unterbricht SOAR die aktuelle Suche und setzt es sich als neues Ziel herauszufinden, welche der Regeln angewendet werden soll. Danach wird die Suche fortgesetzt.

ACT-R löst dieses Problem etwas anders – wie bereits erwähnt werden für alle möglichen Regeln Aktivierungskosten berechnet: liegen diese sehr nahe zusammen, so entscheidet der zufällige Anteil welche Regel letztendlich aktiviert wird.

4.4 Lernen: Chunking

Betrachtet man nun komplexere Probleme und deren Lösungsprozesse, so fällt auf, dass bestimmte, bereits abgeleitete Fakten immer wiederkehren. Das System wiederholt also eine Folge von Regeln, welche bei bestimmten Ausgangsfakten beginnen und in einer Serie von Ableitungen gewisse Endfakten hervorbringen. Der hierfür notwendige Ableitungsprozess wird also ständig wiederholt. Genau diese Tatsache macht sich das Chunking zu Nutze. Es werden dabei sich ständig wiederholende Ableitungspfade ausgewählt, und als abkürzende, direkte Regeln in die prozedurale Wissensbasis aufgenommen.

5. Bekannte Architekturen

Mittlerweile hat sich eine Vielzahl kognitiver Architekturen gebildet. In diesem Kapitel sollen die zwei bekanntesten kurz angesprochen werden.

5.1 SOAR

SOAR (<http://sitemaker.umich.edu/soar/home>) wurden am Anfang von John Laird, Allen Newell, und Paul Rosenbloom an der Carnegie Mellon Universität entwickelt. Vom Beginn an der Entwicklung dieser Software im Jahre 1983 fand sie stets Anwendung in vielen praktischen Versuchen, in denen verschiedene psychologische Theorien modelliert und getestet wurden.

Der ausgeschriebene Name – State Operator Apply Result – betont die vorher besprochene Struktur kognitiver Architekturen. Die Besonderheiten von SOAR liegen z.B. darin, dass der Suchraum durch so genannte Operatorregeln aufgespannt wird. Diese werden zuerst durch Propose-Regeln vorgeschlagen, also auf diese Weise als Operatoren gekennzeichnet.

Ebenfalls einzigartig ist der Aufbau des Speichers. Bei SOAR wird das prozedurale und das faktenbasierte Wissen im selben Arbeitsspeicher abgelegt, dem sogenannten Working Memory. Die Trennung wie bei den meisten anderen

Systemen entfällt. Weiterhin ist bei SOAR der Speicher in Form eines Graphen organisiert. Zu jedem Faktum können so ständig neue Fakten und Attribute hinzugefügt werden. Das „Vergessen“ bzw. Löschen von Informationen wird durch eine Trennung der Fakten von der Wurzel hervorgerufen. Alle unverbundenen Teilgraphen werden von SOAR gelöscht.

```
sp { sayhello* I am, so say hello
    (state <s> ^type state)
    →
    (<s> ^sayit <hw>)
    (<hw> ^say helloworld)
}
sp { propose*helloworld
    (state <s> ^say helloworld)
    →
    (<s> ^operator <o> +)
    (<o> ^name helloworld)
}
sp { apply*helloworld
    (state <s> ^operator <o>)
    (<o> ^name helloworld)
    →
    (write |Hello World!|)
}
```

Abbildung 2: Auszug aus einem SOAR Programm

Abbildung 2 zeigt ein kleines SOAR Programm, welches drei verschiedene Kategorien von Regeln demonstriert. Die erste Regel zeigt, dass in SOAR „statisches“ Wissen durch spezielle Regeln beim Start des Prozesses in das Working Memory übertragen wird. Die zweite Regel schlägt einen Operator zum „Aufspannen“ des Suchraumes vor. Die dritte Regel definiert was bei der Ausführung des Operators passieren soll.

Der SOAR Kern wurde in C entwickelt. Konfiguration und Start passieren durch TCL Interfaces. Mit SML (Soar Markup Language) bietet SOAR in den neuen 8.6.* Versionen ein völlig neues, von Programmiersprachen unabhängiges XML-Interface.

5.2 ACT-R

ACT-R (<http://act-r.psy.cmu.edu/>) ist entstanden aus den Ergebnissen der Forschungsarbeit von John Robert Anderson. Ebenfalls entwickelt an der Carnegie

Mellon Universität in Pittsburgh, nahmen die Forschungsarbeiten von Allen Newell einen starken Einfluss auf die Entwicklung von ACT-R.

Bei ACT-R wird das deklarative Faktenwissen in Form von Chunks in spezielle Speicherbereiche, den so genannte Buffern abgespeichert. In verschiedenen Modulen angeordnet, können so durch die Kombination von entsprechendem Faktenwissen und den dazugehörigen prozeduralen Regeln die Funktionsweise bestimmter Bereiche des Gehirns simuliert werden.

```
(CLEAR-ALL)
(CHUNK-TYPE addition-fact addend1 addend2 sum)
(CHUNK-TYPE integer value)
(ADD-DM (fact3+4
  isa addition-fact
  addend1 three
  addend2 four
  sum seven)
  (three
    isa integer
    value 3)
  (four
    isa integer
    value 4)
  (seven
    isa integer
    value 7))
```

Abbildung 3: Ein Beispiel für Chunks in ACT-R.

Im Gegensatz zu SOAR wird hier deklaratives Wissen direkt in entsprechender Syntax kodiert. Hier werden die Zahlen 3, 4 und 7 eingeführt, deren Typ und die Tatsache, dass die Summe der Zahlen 3 und 4 die Zahl 7 ergibt.

ACT-R wurde in der Programmiersprache LISP entwickelt. Die Software läuft auf allen gängigen Plattformen und ist seit kurzem in der Version 6 verfügbar. Es gibt mittlerweile viele Systeme, die auf ACT-R aufgebaut sind. Mittlerweile ist sogar eine Python Version verfügbar.

6. Unified Theories of Cognition

Angestoßen von Allen Newell [4], entwickelte sich hinter diesem Begriff die Idee, die komplette Bandbreite menschlicher kognitiven Fähigkeiten in einer einzigen, großen Theorie zu vereinigen. Die etwas widersprüchliche Verwendung des Plurals soll hierbei andeuten, dass momentan alle Theorien weit davon entfernt sind, eine vollständige Abdeckung zu erreichen. Daher werden die existierenden Systeme als eine breite Palette an Kandidaten hierfür angesehen. Die Hoffnung besteht also darin, die Abdeckung der einzelnen Systeme durch Weiterentwicklung und gegenseitigen Einfluss zu steigern, um sich so einer vereinheitlichten Theorie anzunähern.

Weiterhin werden in der Publikation von Allen Newell grundlegende Strukturen angesprochen, und deren Existenz als Baustein aller intelligenten Wesen argumentiert. Als ein wichtiger Bestandteil menschlicher Intelligenz wird hier z.B. eine Wissensbasis angesehen. Man sieht den Mensch hier als „Wissenssystem“, welches Informationen über seine Umwelt, seine möglichen Aktionen und seine Ziele hat und dabei anhand des verfügbaren Wissens genau die Operation auswählt, welche seine Ziele erfüllen.

Akzeptiert man nun eine Form von Wissensbasis als Bestandteil jeder Intelligenz, so wird einem bei der Umsetzung am Computermodell schnell eine weitere, unbewusste Fähigkeit unseres Gehirnes klar: Unabhängig davon, welche Information wir in unserem Gehirn abspeichern wollen, das Gehirn ist dazu in Lage jegliche Art von Information mehr oder weniger effizient zu speichern. Egal ob wir versuchen uns bildlich ein Objekt einzuprägen, ob wir versuchen uns die Gesichtszüge einer Person zu merken oder ob wir nur Gerüche oder Töne speichern – wir brauchen uns dabei keine Gedanken darüber zu machen, wie unser Gehirn diese Informationen repräsentiert. Genau dies stellt aber bei der Umsetzung der Modelle auf dem Computer eines der Hauptprobleme dar. Es werden hier also verschiedene Methoden zur Wissensrepräsentation besprochen und dabei betont, dass dieses Gebiet eines der komplexesten Probleme kognitiver Architekturen darstellt.

Als eines der wichtigsten Konzepte der Wissensrepräsentation wird dabei das Symbol angesehen. *“Symbols provide distal access to knowledge-bearing structures that are located physically elsewhere within the system.”* [3, p 427]. Symbole bieten die Möglichkeit, verschiedene Objekte des Denkens miteinander in Verbindung zu bringen.

Da eines der Ziele der Forschung darin besteht, intelligentes Verhalten am Modell nachzubilden, wird natürlich auch ein Versuch unternommen, dieses Konzept zu definieren. Zusammenfassend kann man folgende Aussage erkennen: *„A system is intelligent to the degree that it approximates a knowledge-level system.“* [3, p 428]. Um also intelligent zu handeln, braucht jedes Wesen eine Wissensverwaltung.

7. Zusammenfassung

Die Kognitionswissenschaft unternimmt mit ihren kognitiven Architekturen und deren Modellen den Versuch intelligentes Verhalten am Computer nachzubilden. Dabei will man nicht nur sämtliche kognitiven Fähigkeiten des Menschen am Computer nachbilden, sondern auch grundlegende Aussagen über die Struktur jeglicher Intelligenz werden getroffen. Wichtige Ergebnisse liefern die jahrelangen Forschungen z.B. mit der Erkenntnis, dass jedes intelligente Wesen eine Form von Wissensbasis benötigt, und wie diese ihre Informationen repräsentiert. Manche beim Menschen beobachtete Konzepte, wie z.B. Lang- und Kurzzeitgedächtnis als Speicher für Informationen, werden in den Modellen vereinfacht dargestellt.

Die verschiedenen Architekturen selber besitzen meist eine eigene Sprache, in der durch Regeln und Fakten ein Produktionssystem gesteuert wird, um so neues Wissen und durch entsprechende Sensoren und Aktoren intelligentes Verhalten zu produzieren. Ein Konzept des Lernens, Chunking genannt, generiert im Suchraum Abkürzungen, bei denen häufig reproduziertes Wissen direkt als Produkt seiner Voraussetzungen in der Wissensbasis übernommen wird. Die Suche nach Lösungen in komplexen Suchräumen, welche durch die ausdrucksstarken Sprachen und entsprechenden Problemstellungen hervorgebracht werden können, wird durch spezielle Mechanismen wie Metaregeln und statistisches Lernen beschleunigt.

Mit SOAR und ACT-R als Modelle menschlicher Kognition haben sich zwei umfangreiche Architekturen gebildet, welche mittlerweile in einer Vielzahl von wissenschaftlichen Versuchen benutzt werden, und in die auch das Wissen einer Vielzahl von Erkenntnissen aus der Psychologie, der Kognitionswissenschaften und der Informatik einfließt.

Beim Betrachten des aktuellen Forschungsstandes in Bereich Künstliche Intelligenz erkennt man schnell, dass sich hier keine Kerntheorie erkennen lässt, welche ohne große Abhängigkeiten von anderen Wissenschaften eine breite Palette an Erkenntnissen zulässt. Vielmehr hat sich die Künstliche Intelligenz, früher eher als eine theoretische, eigenständige Disziplin der Informatik angesehen, nun zu einer praktischeren, interdisziplinären Forschung in den Bereichen der Psychologie, der Kognitionswissenschaften und der Informatik entwickelt. Vergleicht man hierbei die Ergebnisse der Forschung aller Bereiche der Künstlichen Intelligenz, so darf man die kognitiven Architekturen, neben den konnektionistischen Ansätzen, als eine der Hauptkandidaten für die Nachbildung intelligenten Verhaltens am Computer ansehen.

8. Literatur

- [1] ACT-R, <http://act-r.psy.cmu.edu/>, 21.03.2007
- [2] Langley, Pat, Laird, John E., Rogers, Seth (Technical Report) Cognitive Architectures: Research Issues and Challenges, Computational Learning Laboratory, CSLI, Stanford University
<http://csl.stanford.edu/~langley/papers/final.arch.pdf>
- [3] Newell, Allen Précis of Unified Theories of cognition. In Behavioral and Brain Sciences 15. pp 425–492
- [4] Newell, Allen, 1990, Unified theories of cognition, Harvard University Press
- [5] SOAR, <http://sitemaker.umich.edu/soar/home>, 21.03.2

