Dirk Rehberger

Das System DENDRAL

Ausarbeitung zu einem Seminarvortrag im Seminar „Wissensbasierte Systeme“, Prof. Dr. M. Schmidt-Schauß, Wintersemester 1995/96, Fachbereich Informatik, JWG-Universität Frankfurt am Main

Was ist DENDRAL? – Einführung **2**
- **Kurzbeschreibung** **2**
- **Allgemeines** **2**
- **Das Wissensprinzip** **2**
- **Geschichtlicher Hintergrund in der KI** **2**
- **Wie es zu DENDRAL kam und die weitere Entwicklung** **3**
- **Chemische Grundlagen** **3**
- **Das fundamentale Problem der Analytischen Chemie – Die Strukturbestimmung von Molekülen** **4**
- **Massenspektrometrie** **4**

Aufbau und Vorgehen von DENDRAL **5**
- **Der Generator** **5**
- **Das Planen** **6**
- **Das Testen** **7**
- Repräsentation von Strukturformeln und Produktionsregeln 7
- Massenspektrometrie-Theorien in DENDRAL 8
- Ranking 9

Resultate und Schlußfolgerungen **9**
- **CONGEN** **9**
- **Status** **9**
- **Lehren für die Implementierung intelligenter Systeme** **10**
- **Organisation des Projektes** **12**
- **Knowledge engineering** **12**
- **Wieviel weiß DENDRAL?** **13**
- Das Wissen von DENDRAL über chemische Konzepte und Prozeduren 13
- **Beobachtungen über wissenschaftliche Entdeckungen** **13**
- **Schlußbemerkungen** **14**

Anlagen
- Tabelle: Die Programme von DENDRAL
- Beispieldurchlauf con CONGEN
- Liste: Das Wissen von DENDRAL über chemische Konzepte und Prozeduren

Was ist DENDRAL? – Einführung

Kurzbeschreibung

DENDRAL ist ein Projekt, das sich mit dem KI-Problem des Verstehens wissenschaftlicher Schlußfolgerungen beschäftigte, im Kontext eines wichtigen und allgegenwärtigen Problems der Organischen Chemie. Konkret geht es bei DENDRAL darum, aus experimentellen Daten (Massenspektrum sowie von Hand eingegebene Constraints) die Molekularstruktur eines Stoffes zu bestimmen. Der Name DENDRAL kommt von DENdritic ALgorithm, einem Algorithmus, der alle topologisch verschiedenen Anordnungen einer gegebenen Menge von Atomen genau einmal aufzählt und das Herz der Programme darstellt.
Die Hauptlehre von DENDRAL ist, daß es möglich ist, Probleme mittlerer Komplexität, etwas reduziert, in ein System zu packen, das dem Menschen beim Problemlösen assistiert. Wenn man nicht versucht, breite, allgemeine Probleme zu lösen, sondern ein spezielles Problem, und man so viel spezifisches Fachwissen einbaut wie möglich, kann ein nützliches System herauskommen.

Allgemeines

DENDRAL ist: Die erste größere Anwendung heuristischer Programmierung auf eine experimentelle Analyse in einer empirischen Wissenschaft; eines der ersten großen Programme, das detailliertes, fachspezifisches Wissen über den Anwendungsbereich für Heuristiken benutzte; und das erste regelbasierte System, das auf ein „real world“-Problem angewandt wurde. Es ist ein Expertensystem, das von Chemikern für ihre eigenen Zwecke benutzt wurde.
Es war eines der größeren KI-Projekte; dabei war es ein interdisziplinäres Projekt, das sich über mehr als ein Jahrzehnt lang entwickelte, und stellte eine ausgedehnte empirische Erforschung heuristischer Programmiertechniken dar. Es hat viele neue Programmorganisationskonzepte eingeführt, die später oft angewandt wurden.
Es ist ein wissensbasiertes System. Hier wurde zuerst das Konzept einer separaten Wissensbasis angewandt, die für sich editiert werden kann, ohne den Rest des Programms, der das Wissen anwendet, zu verändern. Das Wissensprinzip kam zuerst im Kontext von DENDRAL auf: Wissen ist Macht.

Das Wissensprinzip

Das Wissensprinzip besagt, daß Intelligenz und Kompetenz eines Systems hauptsächlich durch Fachwissen über den Problembereich zustande kommt, das im System verankert ist – Konzepte, Fakten, Repräsentierungen, Methoden, Modelle, Metaphern, Heuristiken.
Das Prinzip wird breit akzeptiert. In der frühen Entwicklungsphase von DENDRAL stand es aber in großem Gegensatz zum damals vorherrschenden Standpunkt, der Intelligenz vor allem der Macht des logischen Schließens zuschrieb. Das Wissensprinzip sagt dagegen, daß spezifisches Wissen die Hauptquelle von menschlicher und künstlicher Intelligenz ist und dafür sogar einfache Schlußregeln genügen! Dafür gibt es inzwischen mehr als genug empirische Beispiele, von denen DENDRAL ein wichtiges ist.

Geschichtlicher Hintergrund in der KI

Das Hauptziel der KI-Forschung besteht darin, intelligentes Denken zu verstehen.

Die erste Dekade der KI (ca. 1954-64) war von Enthusiasmus geprägt. Es herrschte Optimismus, daß viele Probleme aus Bereichen wie Psychologie, Linguistik, Mathematik, Philosophie, Technologie und Management bald gelöst werden würden. Die Anfänge von DENDRAL liegen am Ende dieser Dekade. Zu dieser Zeit haben sich Probleme, zu denen optimistische Voraussagen gemacht wurden, als unverhofft schwierig herausgestellt. Klassisches Beispiel hierfür ist die automatische Übersetzung natürlicher Sprachen.
In der Mitte der sechziger Jahre befand sich die KI in einem Übergangsstadium: Der Optimismus war gedämpft, doch für Programme für allgemeine Problemlösung wurde noch viel geforscht; auch für einen GPS – General Problem Solver. Ein positives Ergebnis hierbei war die Entdeckung der Resolutionsprozedur für Beweise in der Prädikatenlogik. Die Computertechnologie schritt zu dieser Zeit rapide voran: Schnellere, verläßlichere Hardware kam auf und fähigere Software, und Telekommunikation über Computer entwickelte sich.
Die erste große Implementation von DENDRAL ging noch über Telefonfernleitungen, die Computer in Palo Alto und Santa Monica miteinander verbanden – Telefonrechnungen von $ 2000 im Monat waren keine Seltenheit. In diesem Milieu fing DENDRAL an.

Wie es zu DENDRAL kam und die weitere Entwicklung

Um 1963 fing Edward A. Feigenbaum an, sich für informationsverarbeitende Modelle menschlicher empirischer Induktion zu interessieren. 1965 erklärte er Joshua Lederberg, daß er ein passendes wissenschaftliches Problem suche, um Induktion von wissenschaftlichen Hypothesen zu studieren, Lederberg schlug die Strukturbestimmung von Molekülen mit Hilfe des Massenspektrums vor. Die (außerhalb der KI liegende) Motivation hierzu war, daß Lederberg in einem Projekt der NASA war, das nach Leben auf dem Mars suchte – mit einer automatisierten Strukturbestimmung könnten auch noch nicht bekannte Stoffe identifiziert werden.
Ein erster Prototyp des Generators mit dem von Lederberg entwickelten „Dendritic Algorithm“ wurde 1965 von William C. White programmiert, und Ende dieses Jahres stand die Definition des Projektes fest. Georgia Sutherland und Bruce G. Buchanan wurden 1966 rekrutiert.
Es wurde viel mit Chemikern zusammengearbeitet (v. a. Carl Djerassi), damit sie ihr Fachwissen einbringen konnten. Das Vorgehen hierbei wurde von der Frage „Was weißt du, was das Programm noch nicht weiß?“ geleitet – ein Grundpfeiler des „knowledge engineering“, des Prozesses der Zusammenarbeit von Programmierern mit Experten des betreffenden Fachgebiets.
DENDRAL behandelte zunächst nur aliphatische (d. h. azyklische) Strukturen. Die Liste der Stoffe, mit denen es zu tun hatte, ist: Aminosäuren, Ketone, Ether, Alkohole, Amine, Thiole und Thiolether, in ungefähr dieser Reihenfolge. Später kamen auch Stoffe mit zyklischen Molekülstrukturen hinzu, nachdem der Dendritic Algorithm von azyklischen auf alle möglichen Strukturen erweitert wurde. Das praktische Limit für die Anzahl der Atome in einem Molekül, die behandelt werden kann, lag wegen der konventionellen Massenspektrometrie bei ca. 100 Atomen. Es gibt inzwischen Massenspektren mit viel höherer Genauigkeit, so daß Strukturen mit Tausenden von Atomen behandelt werden können, doch da es diese noch nicht lange gibt, wurde DENDRAL (noch) nicht darauf angewandt.
Nachdem das System anfangs von denjenigen Projektmitgliedern entwickelt worden war, die an der Informatik orientiert waren, arbeiteten später hauptsächlich computerorientierte Chemiker daran, im Sinn, ein praktischeres und benutzerfreundlicheres System aus DENDRAL zu machen. Nach mehr als 15 Jahren intensiver Arbeit wurde DENDRAL lizenziert und die Lizenz an einen Konzern vergeben, der Software an Chemiker verkauft.

Chemische Grundlagen

Ein Stoff besteht aus gleichen Molekülen. Jedes Molekül setzt sich aus Atomen (in der organischen Chemie meistens Atome der Elemente Kohlenstoff (C), Wasserstoff (H), Stickstoff (N), Sauerstoff (O), Phosphor (P) und Schwefel (S)) zusammen, die in einer stabilen dreidimensionalen Struktur durch

gemeinsame Benutzung von Elektronenpaaren gebunden sind. Solches „electron sharing“ zwischen zwei Atomen nennt man eine chemische Bindung. Es gibt Einfach-, Doppel- und Dreifachbindungen. Die **Summenformel** eines Moleküls gibt an, wie viele Atome von jeder Art im Molekül enthalten sind, z. B. $C_8H_{16}O_2$. Ein Molekül kann man schematisch als ungerichteten, markierten Graphen darstellen (**Strukturformel**), dessen Knoten die Atome sind und dessen Kanten die Bindungen: z. B.

Dabei kann ein Knoten des Graphen auch für eine ganze Teilstruktur stehen, die dann durch ihre Summenformel angegeben ist.
In der Organischen Chemie besitzen Moleküle grundsätzlich ein Gerüst, das sich aus Kohlenstoffatomen zusammensetzt; diese haben immer jeweils 4 Bindungen („Kohlenstoff ist vierwertig“). Meist wird im Graphen das „C“ für Kohlenstoff weggelassen und mit ihm die H-Atome, die an diesem C-Atom hängen.
Man unterscheidet zwischen der **Topologie** und der **Geometrie** der Moleküle; ersteres ist der schematische Aufbau (wie im Graphen), letzteres bedeutet die Topologie zusammen mit Bindungswinkeln und -längen. Zwei verschiedene Moleküle heißen **Isomere,** wenn sie die gleiche Summenformel haben. Besitzen sie auch noch die gleiche Topologie, so sind sie **Stereoisomere**. (Z. B. kann ein C-Atom, das mit vier verschiedenen Atomgruppen verbunden ist, das auf zwei geometrisch verschiedene (zumindest im dreidimensionalen Raum), topologisch gleiche Weisen sein.)
Ein **Ion** ist ein elektrisch geladenes Molekül.

Das fundamentale Problem der Analytischen Chemie – Die Strukturbestimmung von Molekülen

Dies ist das Problem, mit dem DENDRAL sich beschäftigt. Es ist ein wichtiges, grundlegendes Problem in der Organischen Chemie, denn chemische und physikalische Eigenschaften eines Stoffes hängen – außer Anzahl und Art der Atome – davon ab, wie die Organisation der Atome im Molekül aussieht.
Zur Strukturbestimmung von Molekülen gibt es verschiedene Mittel. Eines der fruchtbarsten und am häufigsten benutzten ist die **Massenspektrometrie**, hierzu gleich mehr. DENDRAL benutzte zuerst nur Informationen aus dieser Methode und behandelte daher verschiedene Molekülgeometrien gleich (später wurde Wissen über dreidimensionale Strukturen hinzugefügt). Andere Methoden sind z. B. Gaschromatographie, Infrarot- und Ultraviolettspektrometrie, Kernresonanzspektroskopie. Ergebnisse aus diesen Methoden können DENDRAL durch manuelle Eingabe von Constraints mitgeteilt werden; im Laufe der Zeit wurden noch Wissen und Programmteile hinzugefügt, die solche Constraints automatisch herleiten können, sowie Wissen und Programme für den Postprozessor (Planer), damit dieser experimentelle Daten auch für diese Methoden vorhersagen kann, um die Kandidatenmoleküle zu unterscheiden.

Massenspektrometrie

In einem **Massenspektrometer** wird der zu untersuchende Stoff mit Elektronen bombardiert. Dadurch zerbrechen die Moleküle in Fragmente, die selbst noch weiter zerfallen können. Manche Bindungen im Molekül zerbrechen einfacher als andere (meistens brechen Einfachbindungen; eine Doppelbindung C=O bleibt nahezu immer erhalten). Die Fragmente sind elektrisch geladen (Ionen). Diese Ionen werden in einem elektrischen Feld beschleunigt und dadurch nach der Masse getrennt; in einem Magnetfeld werden sie abgelenkt und dadurch nach Ladung getrennt. Insgesamt ergibt sich eine Aufteilung der Fragmente nach Masse pro Ladung. Ersichtlich ist das im Ergebnis, im **Massenspektrum**: Auf der Masse/Ladung-Achse ist eine Kurve aufgetragen, die an den Stellen, wo die Fragmente auftreten, „Peaks“ hat, deren Höhe der Häufigkeit der Ionen entspricht.

Aufbau und Vorgehen von DENDRAL

DENDRAL ist nicht ein einzelnes Programm, sondern ein System, das aus mehreren Programmen besteht. Eigentlich sind es sogar zwei Systeme: Das eine heißt Heuristic DENDRAL. Es beinhaltet Fachwissen aus der Chemie (insbesondere der Massenspektrometrie), akzeptiert ein Massenspektrum und andere Experimentaldaten als Eingabe und gibt eine geordnete Menge chemischer Strukturen (bzw. Beschreibungen derselben) aus, die als Hypothesen für den unbekannten Stoff dienen. Das andere, Meta-DENDRAL, nimmt bekannte Massenspektrum-Struktur-Paare als Eingabe an und versucht, Wissen über Massenspektrometrie zu folgern, das von Heuristic DENDRAL angewendet werden kann. In diesem Papier geht es nur um das erstere System.

Es gibt ein generate-and-test-Programmier-Paradigma: Ein Generator zählt potentielle Lösungen auf, die durch einen Tester auf Plausibilität überprüft werden. Der Aufbau von DENDRAL folgt einer Erweiterung dieses Paradigmas: Es hat eine **plan-generate-test**-Struktur. Dabei gibt ein Planer dem Generator Constraints mit auf den Weg, wodurch gewisse unmögliche Hypothesen von vornherein ausgeschlossen werden.
Bei einem Generator ist erwünscht, daß er alle möglichen Lösungen ausgibt („erschöpfend“), aber jede nur einmal (wegen der Effizienz). Der Planer soll sehr flexibel sein und neues Wissen leicht annehmen können. Das Wissen ist hier deshalb modular angelegt: Man kann neues Wissen einfach zum Programm hinzufügen, ohne das schon vorhandene Wissen irgendwo verändern zu müssen.
Die Arbeit des Testers ist es, zu jeder vom Generator vorgeschlagenen Lösung ein passendes Massenspektrum vorherzusagen und mit dem ursprünglichen Labor-Spektrum zu vergleichen.
Planer und Tester verringern beide die Menge der möglichen Lösungshypothesen, wobei sie mit Daten (Massenspektrum) und Wissen (Theorie der Massenspektrometrie) operieren. Es ist nun im Prinzip viel ökonomischer, diese Verringerung in der Planphase vorzunehmen, vor der Generierung, als danach beim Testen. Deswegen sollte man auch so viel Wissen und so viele Constraints wie möglich in die Planphase ziehen, damit bereits die Generierung eingeschränkt wird. Das ist jedoch nicht immer möglich (s. u., Tester).
Damit die drei Komponenten Planen, Generieren, Testen überhaupt gut zusammenarbeiten können, braucht man eine uniforme Darstellung der Datenobjekte in allen drei Teilen. Bei DENDRAL sind es die Strukturformeln (Graphen), die die Moleküle repräsentieren. Alle Komponenten des Systems arbeiten mit dieser Darstellung. Diese gemeinsame Nutzung der Repräsentation ist gewissermaßen der Kleber, der die drei Teile zusammenhält. (Nebenbei können Strukturformeln als Listen in LISP gut dargestellt werden. Wie, dazu später mehr.)

(Tabelle: Die Programme von DENDRAL)

Der Generator

Der Generator ist in der Hauptsache das Programm CONGEN. Es zählt alle möglichen Graphen (bzw. Strukturen) auf, die eine gegebene Zahl von Knoten (bzw. Atomen) von gegebenem Typ (bzw. Element) haben. Da der Algorithmus unabhängig von der chemischen Interpretation ist, kann er auch für andere Systeme verwendet werden, die ihre Hypothesen als markierte Graphen darstellen. (Den Autoren von [1] sind allerdings noch keine wirklich benutzten Anwendungen bekannt.) Man kann CONGEN vielerlei Constraints mitgeben (CONGEN ist „Constrained Generation“), es ist somit ein gutes Beispiel für das Konzept der heuristischen Suche in der Menge der möglichen Lösungen. Erlaubte Constraints sind: (1) Unterstrukturen, die in bestimmter Anzahl vorhanden sein müssen, (2) Größe von Ringen (Zyklen), die vorhanden sein müssen, (3) Anzahl von H-Atomen, die an einer bestimmten Struktur vorhanden sein müssen, und (4) Anzahl der Isopreneinheiten, die vorhanden sein müssen. (Das sind fünfelementige Ketten von Kohlenstoffatomen, die Y-förmig verbunden sind.)
Diese Constraints sind flexibel und vielfältig, siehe dazu den Beispieldurchlauf von CONGEN.

Als erster Arbeitsschritt ist es möglich, „Superatome“ zu definieren, d. h. zusammenhängende Teilgraphen, die beim Generieren zunächst als einzelne Knoten betrachtet werden. Diese können später „ausgepackt“ werden.
CONGEN arbeitet interaktiv; zu jedem Zeitpunkt können vom Benutzer Constraints eingegeben werden. Das ist z. B. nützlich, wenn man merkt, daß zu viele mögliche Lösungen generiert werden und man die Möglichkeiten noch weiter einschränken will. Die Suche nach den Kandidaten ist eine Tiefensuche; so kann man sich zu jeder Zeit über die Anzahl schon gefundener Lösungen informieren. Außerdem kann so eine Schätzung der Anzahl verbleibender Strukturen auf Verlangen ausgegeben werden.

(siehe Beispieldurchlauf von CONGEN)

Das Programm GENOA ist eine Weiterentwicklung von CONGEN, die nicht von Anfang an bei DENDRAL dabei war. Es bietet diverse Erleichterungen und Komfort für den Benutzer:
- Zwischenergebnisse bei der Erzeugung können untersucht werden (d. h. innere Knoten des Generierungsbaums, oder Mengen von Lösungskandidaten).
- Mehrere benutzerdefinierte Unterstrukturen (Superatome) dürfen in den Ergebnismolekülen überlappen. (GENOA = „Generation With Overlapping Atoms“)
- Bei Unterstrukturen können Alternativen definiert werden, d. h. es ist möglich zu sagen, daß von bestimmten Teilstrukturen eine, aber nicht alle vorhanden sein sollen.

Die beiden letzten Punkte lassen es zu, daß nun ein Constraint ausreicht bei Fällen, die unter CONGEN mehrere Constraints brauchen würden. GENOA behandelt diese Fälle, indem es sie selbst in die verschiedenen Teilhypothesen aufspaltet, aus denen sie bestehen und die von CONGEN bearbeitet werden können.
CONGEN und GENOA können nur Isomere generieren, die sich in der Topologie unterscheiden. Aus diesen können vom Programm STEREO noch Stereoisomere erzeugt werden. Weitere Programme wurden geschrieben (v.a. von James Nourse), die mit der dreidimensionalen Struktur von Molekülen noch besser umgehen können (Anzeige; Formulierung von Constraints bezüglich der 3D-Struktur).

Das Planen

Beim Generator werden die Constraints manuell vom Anwender eingegeben. In der Planphase (hauptsächlich das Programm PLANNER) werden sie von DENDRAL selbst erzeugt. Die erste Version des Planers untersuchte das Massenspektrum und schloß mit eingebauten Regeln, die von einem Chemiker erstellt wurden, auf Strukturen, die vorhanden sein müssen. In späteren Versionen wurden diese Regeln selbst von einem Regelgenerator erzeugt. Wichtig ist, daß das Wissen im PLANNER explizit und austauschbar dargestellt ist, so daß Regeln leicht hinzugefügt oder verbessert werden können. Im Laufe der Zeit wurde auch klassenspezifisches Wissen, d. h. Wissen über einzelne Stoffklassen, hinzugefügt.
Der Input der Planphase ist ein Grundskelett der Stoffklasse, die vorliegt; Definitionen der verschiedenen möglichen Bruchstellen (beim Zerfallen im Massenspektrometer) sowie das gemessene Massenspektrum. Das Hilfsprogramm MOLION kann gewöhnlich aus dem Massenspektrum bereits die Summenformel des Stoffs ermitteln. PLANNER formuliert aus diesen Daten Constraints, die angeben, an welchen Stellen des Grundskeletts wahrscheinlich die Substituenten hängen und wo wahrscheinlich nicht.
PLANNER enthält Regeln von der Form: Daten → Situation (in der diese Daten produziert werden), nämlich Menge von m/e-peaks → Unterstruktur des Moleküls. Auch prädiktive Regeln sind vorhanden. Sie haben die Form Situation → Aktion, nämlich Teilgraph → massenspektrometischer Vorgang (Fragmentierungen und Umstrukturierungen). Dazu muß das Grundgerüst bekannt sein!
Mit Summenformel und Grundskelett können Anzahl und Typ der Atome und der Mehrfachbindungen ermittelt werden. Es bleibt noch herauszufinden, wie die Substituenten (das, was am Grundskelett dranhängt) genau aussehen und wo sie sitzen. Durch die prädiktiven Regeln werden von verschiedenen Situationen aus Peaks ermittelt, die man sehen würde, wenn die gegebene Situation vorhanden ist, und

die mit dem originalen Spektrum verglichen werden. Damit wird der Kreis der Lösungskandidaten weiter eingeschränkt.

Das Testen

Das Testen wird erst dadurch überhaupt notwendig, daß das Planen nicht perfekt ist. Dies ist vor allem auf folgende Gründe zurückzuführen: Manche Tests können nur mit voll spezifizierten Kandidaten durchgeführt werden. Außerdem kann es sein, daß Tests in der Planphase zu teuer sind und erst nach dem Generierungsschritt effizient durchzuführen sind.
Das Hauptprogramm beim Testen ist PREDICTOR. Es besteht hauptsächlich aus einer Menge von Produktionsregeln; sie definieren eine Theorie des Verhaltens eines Stoffes im Massenspektrometer: „Strukturen einer gegebenen Form werden auf diese und jene Weise zerfallen." Diese Regeln werden auf jede Kandidatenstruktur angewandt, so daß jeweils ein Massenspektrum hervorgesagt wird, das mit den ursprünglichen Experimentaldaten verglichen werden kann. Die Regeln sind im Prinzip wie die prädiktiven Regeln im Planer, nur allgemeiner. Je besser das hervorgesagte Massenspektrum mit dem gemessenen übereinstimmt, desto höher ist die Priorität, die die betreffende Molekülstruktur für die Ausgabe erhält.
Die Produktionsregeln haben wieder das Format Situation → Aktion. Die „Situation" ist die Beschreibung einer Strukturformel, die „Aktion" besteht aus Operationen, die diese Strukturformel verändern. Dabei kann eine Aktion auch eine ganze Produktion enthalten.
Das Vorgehen von PREDICTOR ist dieses: Zuerst wird aus dem gegebenen Molekül ein molekulares Ion erzeugt, d. h. eines, das genauso aussieht wie das Molekül, nur ist es einfach positiv geladen. Dieses Ion kommt nun in die – zunächst leere – Ionenliste. Auf ein Ion in der Ionenliste wird jede darauf anwendbare Produktion angewandt; daraus resultieren womöglich weitere Ionen, die hinten an die Liste angefügt werden, und das verarbeitete Ion kommt auf die – auch anfangs leere – Spektrumsliste. Wenn die Ionenliste leer ist, entspricht die Spektrumsliste dem vorhergesagten Massenspektrum: Eine Liste von Ionen mit Masse (pro Ladung) und relativer Häufigkeit für jedes Ion. Man beachte, daß es möglich ist, das gleiche Ionen durch unterschiedliche Zerfallsvorgänge entstehen.

Repräsentation von Strukturformeln und Produktionsregeln

Strukturformeln: Jedes Atom bekommt eine Nummer und einen Namen. Am besten ist wohl ein Beispiel:

```
(PHENOL   (C1 C 1 (2 3 7) 1 0)
          (O2 O 2 (1)     0 1)
          (C3 C 3 (1 4)   1 1)
          (C4 C 4 (3 5)   1 1)
          (C5 C 5 (4 6)   1 1)
          (C6 C 6 (5 7)   1 1)
          (C7 C 7 (1 6)   1 1)
```

Name | Typ (Element) | Nummer | Liste der Nachbarn (ohne H's) | dots | Anzahl benachbarter H's

(„dots" bezeichnet die Zahl der Mehrfachbindungen am Atom.)
Ein Beispiel für Unterstrukturen:

```
(ALCOHOL  (C1 C 1 (2 x -) -)
          (O2 O 2 (1)     0 1)
```

Hierbei heißt „x", daß eine Bindung zu einem Atom außerhalb der Struktur vorliegt. „-" ist ein „don't care"-Eintrag.

Produktionsregeln: Der Situations-Teil kann folgendes sein:

`NIL`	ist immer falsch
`DEFAULT`	wird genau dann wahr, wenn noch keine Produktion angewendet werden konnte
`ISIT(X)`	ist Unterstruktur X im untersuchten Ion vorhanden?
`DB(N,M)`	Besteht eine Doppelbindung zwischen N und M?
`MISSING(N,M)`	sind N und M nicht miteinander verbunden?
`ON(S,N)`	hängt Unterstruktur S am Atom N?
`WHERE(S)`	wie ISIT, nur daß im Erfolgsfall eine Liste ausgegeben wird, die angibt, welche Atome im untersuchten Ion welchen Atomen von S entsprechen, und die vom Aktions-Teil benutzt werden kann (das geschieht übrigens auch bei ON)

Oder jede andere LISP-Funktion.

Der Aktions-Teil ist eine Liste von Komponenten, die entweder von der Form

(LABEL; BREAK-FUNCTION; INTENSITY-FUNCTION; TRANSFER-FUNCTION (optional))

LABEL: zur Kennzeichnung
BREAK-FUNCTION: gibt an, wo die Struktur zerfällt
INTENSITY-FUNCTION: gibt an, mit welcher Häufigkeit der Bruch stattfindet
TRANSFER-FUNCTION (optional): findet Verwendung, wenn Transfer von Wasserstoff (oder anderen Radikalen) stattfindet

sind oder eine ganze Produktionsregel oder eine LISP-Funktion.

Hier ein Beispiel für eine Produktionsregel:

```
((WHERE ESTROGEN)
   (B (BREAKBND ((14. 15) (13.17)))          100 (HTRANS -1 0))
   (D (BREAKBND ((9. 11) (14. 13) (16. 17))) 100 (HTRANS -2 -1))
   (C (BREAKBND ((9. 11) (14. 13) (15. 16))) 100 (HTRANS -1 -0))
   (E (BREAKBND ((11. 12) (8. 14)))          100 (HTRANS -1 0))
   (F (BREAKBND ((9. 11) (8. 14)))           100 (HTRANS -1 0))
```

BREAKBND ist eine bestimmte Break-Funktion, die an den angegebenen Stellen (die sich auf die Atome der Estrogen-Unterstruktur beziehen, die von WHERE zurückgegeben wurden) Fragmentierungen vornimmt. HTRANS ist eine Transfer-Funktion, die Wasserstoff transportiert. Die Intensitäts-Funktion ist hier die Zahl 100; sie bedeutet, daß jedes erzeugt Ion 100 % der Intensität des Ions bekommt, aus dem es hervorgegangen ist.

Massenspektrometrie-Theorien in DENDRAL

Für die Aminosäuren, die ersten Stoffe, für die DENDRAL angewandt wurde, wurde noch die Massenspektrometrie-Theorie „nullter Ordnung“ benutzt: Jede Atombindung kann brechen, und sie tun es auch, und zwar alle nacheinander. Für Aminosäuren funktioniert diese Theorie noch ganz gut, für andere Stoffe allerdings kaum. Sie ist für fast jedes aufkommende Problem viel zu allgemein. Später, in den frühen Siebzigern, baute man klassenspezifische Produktionsregeln ein, die also jeweils nur für einzelne Stoffklassen angewandt werden. Noch später hatte man MSPRUNE (dazu gleich mehr) sowie REACT, das für die Kandidaten Voraussagen bezüglich Reaktionen mit anderen Stoffen trifft, d. h. also ein Programm der Testphase, das nicht mit Massenspektrometrie, sondern mit anderen Mitteln der Chemie umgeht.
MSPRUNE ist die Implementierung einer vereinfachten Form der Halbordnungs-Theorie. Diese wird wieder allgemein angewandt (nicht klassenspezifisch) und ist eine „constrainte“ Version der Theorie nullter Ordnung: Jede Bindung kann brechen – unter den aufgeführten Voraussetzungen. Hier werden die plausibleren von den möglichen Zerfallsprozessen ausgewählt. Die Theorie ist allerdings immer noch keine umfassende Zerfallstheorie.
Zu den Constraints gehören:

- Fragmentierungen erlauben, die durch Bruch von bis zu drei Bindungen entstehen (das ist sinnvoll, in Ringsystemen z. B. kann so ein einzelnes Stück herausbrechen), jedoch keine Doppel-, Dreifach- oder aromatische Bindung trennen.
- Verhindern, daß zwei Bindungsbrüche beim selben C-Atom stattfinden (das wäre energetisch ungünstig).
- H-Transfers auf höchstens 2 Male beschränken.

Ranking

Als letztes ist noch MSRANK zu erwähnen. Das Programm ist dafür zuständig, die vorgeschlagenen Kandidaten danach zu sortieren, wie gut ihr vorhergesagtes Massenspektrum mit den Experimentaldaten übereinstimmt. (Dabei wird den peaks aus dem Spektrum eine Wichtigkeit zugeordnet, die von Masse und Intensität abhängt.) Wie die Funktion aussieht, die die „Güte“ berechnet, nach der sortiert wird, kann hier der Benutzer bestimmen, ebenso wie die Constraints für die Halbordnungs-Theorie.

Resultate und Schlußfolgerungen

CONGEN

CONGEN/GENOA ist von allen Programmen des Systems sicherlich das für allgemeine Zwecke brauchbarste, es wurde von mehreren Chemikern benutzt. Es ist für viele Zwecke anwendbar. Wofür es angewandt wurde, hat die Projektgruppe selbst aber nicht in allen Fällen verfolgt. Es funktioniert für mittelgroße Probleme; mit großen Superatomen können Strukturen mit ≈100 Atomen behandelt werden. Beispiele von Problemen, auf die CONGEN angewandt wurde, sind:

- Chemische Komponenten von Körperflüssigkeiten (z. B. organische Säuren, Aminosäuren) (in der Medizin eingesetzt)
- Terpenoide: C_{15}- und C_{20}-Verbindungen aus natürlichen Ressourcen identifizieren
- Marin-Sterole
- Insektensekrete (zur Identifikation von Hormonen usw.)
- Welche Möglichkeiten der Umbildung gibt es bei ungesättigten und polyzyklischen Molekülen?
- Ionenstrukturen (Gas-Ionen)
- Pharmakologische Reagenzien.

Status

Zumindest bei den Arten Verbindungen, bei denen DENDRAL viel Wissen besitzt, können die Programme mit Laienchemikern mithalten. Doch hat die organische Chemie insgesamt nicht viel von DENDRAL gespürt. Nur wenige Laboratorien haben es benutzt, und wahrscheinlich kennen es überhaupt nur wenige Chemiker.
Eine C-Version von GENOA ist auch kommerziell verfügbar, wurde aber lange nicht weiterentwickelt. Andere Teile von DENDRAL, z. B. ein Algorithmus, der nach gegebenen Unterstrukturen sucht, werden kommerziell genutzt.
Ein Problem ist, daß das System am besten mit kleineren Aufgaben zurechtkommt, für die Chemie-Experten weniger Hilfe brauchen. Größere Probleme, mit denen DENDRAL nicht zurechtkommt, können von Chemikern zu kleineren, Papier-und-Bleistift-Problemen reduziert werden.

Expertensysteme werden natürlich vor allem dazu gemacht, Nicht-Experten zu unterstützen. Doch bei DENDRAL scheint das nicht so sinnvoll zu sein, da die schwierigen Strukturbestimmungsprobleme von den wirklichen Experten angegangen werden und die einfacheren Probleme häufig mit anderen Methoden (z. B. Literaturrecherche) gelöst werden können. Für schwierigste Probleme, für die Experten Unterstützung wollen, ist DENDRAL nicht brauchbar.
Allerdings kann es für eine Ergebniskontrolle nützlich sein, d. h. einerseits die vom Chemiker gefundenen Lösungen zu überprüfen und andererseits von ihm nicht gefundene Lösungen zu entdecken. Findet das System Lösungen, die der Chemiker nicht hat, sind das jedoch meistens (nach einer informellen Studie über ca. 12 veröffentliche Papers) falsche Lösungen, die aber nur deswegen falsch sind, weil Bedingungen gelten, die nicht explizit DENDRAL mitgeteilt wurden und vielleicht in seiner Constraintsprache schlecht formulierbar sind.
Ist die Anzahl der Lösungskandidaten, die DENDRAL liefert, nur gering, kommt man wahrscheinlich sowieso ohne Hilfe des Programms aus. Sind es über 1000, bringt das so gut wie gar nichts. Nur wenn sie in einem schmalen Bereich von etwa 12 bis 100 liegt, ist der Programmlauf überhaupt sinnvoll.
Eine Stärke von DENDRAL ist, daß zu jedem Zeitpunkt auf eine Menge von Kandidaten weitere, neue Constraints – vielleicht auch hypothetische – angewandt werden können, um die Lösungsmenge weiter einzuschränken, ohne die Generierung noch einmal von vorne zu beginnen. Warum sich das nicht vermarkten ließ, ist den Autoren von [1] unbekannt. Mit den Constraints kann man auch Was-wäre-wenn-Spiele machen, um zu schauen, welche chemischen Untersuchungen man als nächste durchführen müßte, um dadurch die Menge der Kandidaten möglichst weit schrumpfen zu lassen. Doch in diesem Punkt ist DENDRAL nicht sehr benutzerfreundlich.
Einige Spekulationen, warum DENDRAL nicht in größerem Umfang benutzt wird:

- Chemiker kennen es nicht
- Es ist zu teuer (Hard- und Software)
- Chemiker möchten die Zeit nicht investieren, es zu lernen
- Es wird nicht als wesentlich angesehen, daß der Generator erschöpfend ist und alle Möglichkeiten erzeugt
- Einen Einstellung der Art „Auf solche Weise wird das nicht gemacht“ oder „Maschinen sollten nicht für mich denken, das mache ich selber“
- Sein Wert wird unterschätzt, weil Benutzer nur wenig mit DENDRAL zusammenarbeiten (da sie die meiste Zeit im Labor tätig sind)
- Für Einzelpersonen lohnt sich das System nicht
- Chemische und pharmazeutische Konzerne arbeiten nicht zusammen (und teilen die Ressourcen)
- Teile des Systems vermarkten sich leichter als das ganze System
- Was DENDRAL tut, wird als zu unwichtig angesehen, um das System zu benutzen.

Lehren für die Implementierung intelligenter Systeme

Als eines der ersten wissensbasierten Systeme ist DENDRAL historisch bedeutsam. Es wurde nach dem Ausschlußparadigma des Problemlösens geschrieben – alle unplausiblen Kandidaten eines vollständigen Lösungsraums werden ausgeschlossen, so daß nur die plausiblen Lösungen übrigbleiben. Es ist mit dem plan-generate-test-Paradigma implementiert; die gute Performance von DENDRAL illustriert die Kraft dieses Paradigmas. Dabei ist die Verwendung eines Generators für DENDRAL eine Notwendigkeit, da eine vollständige, nicht-redundante Aufzählung von verschiedenen chemischen Strukturen gebraucht wird.
Einige Lehren, die man aus der Entwicklung von DENDRAL ziehen könnte:

1– Sehr wichtig ist die Effizienz des Generators, auch wenn Plan- und Testphase effizient sind. Vor allem die Constraints müssen effizient angewandt werden können. Dazu ist zu bemerken, daß ein Constraint-Interpreter entwickelt wurde, der die eingegebenen Constraints in äquivalente, für den Generator möglichst effiziente Constraints umformt. Damit kann im Generierungsschritt viel Zeit gespart werden.

2– Bei der Kandidatensuche ist Tiefensuche (DFS) besser als Breitensuche (BFS). Bei DFS kommen die Endergebnisse nämlich nach und nach zum Vorschein, bei BFS erst am Ende. So kann schneller überprüft werden, ob auch wirklich das Gewünschte berechnet wird. Aber beide Methoden lassen sich gleich gut implementieren!
3– Das Planen ist für die Lösung schwieriger Probleme wesentlich. So viel Wissen wie möglich sollte hier hineingepackt werden, denn hier wird die Suche beschnitten.
4– Es lohnt sich, das Programm flexibel zu halten. So viele Optionen, wie einem einfallen, sollten für den Benutzer eingebaut werden, damit es offen für jede gewünschte Art von Untersuchung wird. Denn jeder Parameter, der fest im Programm kodiert ist, ist eine Einschränkung, die schwierig zu verändern ist und eventuell in Vergessenheit geraten kann.
5– Eine interaktive Brücke (Schnittstelle) zum Benutzer ist wesentlich. Sie sollte knapp, aber informativ sein, sachlich, aber nicht unhöflich.
6– Das plan-generate-test-Paradigma könnte mächtiger werden, wenn man den Generator mit einer Suche kombiniert. Das hieße im Fall von DENDRAL, daß es eine Suche in der Menge der chemischen Strukturen gäbe von der Form, daß bei einem Kandidaten Änderungen vorgenommen werden und geschaut, welche dieser Änderungen das vorhergesagte Massenspektrum möglichst gut an das gemessene angleichen, d. h. die Differenz zwischen den beiden Spektren möglichst klein machen (Ähnlichkeit zur Adaptation beim fallbasierten Schließen). Den Programmierern von DENDRAL fehlten allerdings solche differenzreduzierende Programme; außerdem können kleine Unterschiede in der Molekülstruktur schon große Änderungen im Massenspektrum bewirken.
7– Die Wahl der Programmiersprache ist heutzutage kein großes Thema mehr. Die weitgehende Vernetzung sorgt dafür, daß Programme weit verbreitet werden können. Außerdem ist auch LISP inzwischen sowohl effizienter als auch verbreiteter geworden.
8– Es ist ein realistischeres Ziel, mit dem intelligenten System Problemlösern zu assistieren, als ihre Aufgaben ganz zu übernehmen. Letzteres würde in der Entwicklung außerdem sehr viel länger dauern.
9– Wichtig für das Problemlösen ist es, Programmläufe abzuspeichern, so wie Menschen auch Aufzeichnungen machen und aufbewahren. In DENDRAL sind drei Arten der Speicherung implementiert: Man kann a) Anfangsbedingungen, Zwischen- und Endergebnisse, b) die gesamte Interaktion zwischen Chemiker und Programm, oder c) ein Trace der einzelnen Ablaufschritte des Programms abspeichern.
10– Um ein Programm intelligent zu benutzen, muß der Anwender das Anwendungsgebiet (scope) und die Einschränkungen des Programms verstehen. Gemeint ist folgendes: Z. B. Polymere aufzuzählen, geht mit CONGEN einfach nicht, es liegt außerhalb des Anwendungsgebietes von DENDRAL. Wie aromatische Ringe im Programm definiert sind, ist hingegen eine Einschränkung, die einfacher geändert werden kann
11– Der Kontext, unter dem eine Lösung gefunden wird, d. h. alle eingegebenen und nicht eingegebenen Bedingungen, ist für das Verständnis der Lösung wichtig. Das läuft unter anderem auch auf Punkt 9 hinaus.
12– Uniformität in der Repräsentation von Datenstrukturen (DENDRAL: Strukturformeln) und vom Wissen, das mit diesen Datenstrukturen arbeitet (DENDRAL: Produktionsregeln), fördert die Klarheit des Programms, den Datenfluß und die Programmentwicklung. In DENDRAL ist es der Kleber, der die Programme zusammenhält.

Eine allgemeine Zusammenfassung:
Das erfolgreiche Problemlösen in nichttrivialen Aufgabenbereichen

- benötigt überraschend viel spezielles und allgemeines Wissen,
- benötigt die produktive Zusammenarbeit dieses Wissens, nicht nur dessen Anhäufung,
- benötigt für verschiedene Aufgaben einen unterschiedlichen Organisationsaufbau und
- kann Gewinn schlagen aus einer Art der Repräsentation, die die Verarbeitung im Programm unterstützt.

Organisation des Projektes

DENDRAL ist wohl eines der wenigen erfolgreichen, für lange Zeit interdisziplinären Projekte der Wissenschaft. Mehr als eine Dekade lang wurde zusammengearbeitet; Forscher aus der Chemie, Informatik, Genetik, Philosophie, Physik, Mathematik, Elektroingenieurwesen, Management und Psychologie waren daran beteiligt. Wichtige, wegbereitende Fakten sind die folgenden:
Die Aufgabe sprach viele Interessen an. Es ging nicht um Hypothesenbildung allein, auch nicht um Massenspektroskopie allein.
Die Aufgabe ist nicht unbeschreiblich schwierig, sondern kann von jedem, der einen gewissen Hintergrund hat, verstanden werden.
Ganz wichtig war die Zusammenarbeit: Die Menschen haben sich koordiniert, Streitigkeiten geklärt, die Ideen der anderen respektiert. Die Projektleiter waren Manager, die gelernt hatten, Verantwortung auf die Menschen aufzuteilen. Die Informatiker wollten ein Programm schreiben, das für Chemiker wirklich nützlich ist. Schließlich war es den Mitarbeitern wichtig, mehr als nur halb zu verstehen, was die anderen machten; zumindest denjenigen Mitarbeitern, die nach einiger Zeit als Folge einer natürlichen „Selektion“ übrigblieben.
Man sollte an dieser Stelle noch erwähnen, daß auch ein Sponsor gefunden wurde. Das war nicht einfach, aber es war wichtig, um die Kontinuität des Projektes abzusichern.

Knowledge engineering

Hierbei geht es um die Frage: „Wie gewinne, repräsentiere und benutze ich Wissen, um ein bestimmtes Problem zu lösen?“ Die Entwicklung von DENDRAL lehrt, daß man sich darum gut kümmern muß, um das Wissen produktiv zur Problemlösung einzusetzen.
Zu Beginn des DENDRAL-Projektes bemühte sich die KI-Forschung hauptsächlich um allgemeine Methoden des Problemlösens. Musterbeispiele hierfür sind die Forschung über die Resolutionsmethode für Beweise in der Prädikatenlogik, die viele verschiedene Anwendungen hatte, und die Forschung nach einem „General Problem Solver“. Relativ wenig Arbeit wurde auf den Bereich des Entwurfs von Systemen verwendet, die Fachwissen benutzten. Es gab Ausnahmen, wie Schach-, Dame- oder Programme für symbolische Mathematik.
Heutzutage ist es anders; viele bedeutende Projekte drehen sich um Entwicklung oder Anwendung von wissensbasierten Systemen. Dabei, daß sich diese neue Betonung in der KI etabliert hat, spielte DENDRAL wahrscheinlich eine wichtige Rolle.
Es ist nun (nach Meinung der Autoren von [1]) selbstverständlich, daß zum Problemlösen in einem speziellen Aufgabenbereich auch spezielles Wissen dieses Bereiches benötigt wird. In der frühen KI-Entwicklung war das auch durchaus nicht umstritten. Man war sich nur über die große Menge und Vielfalt dieses Wissens nicht im klaren. Das wurde schließlich bemerkbar, als man sich, nach hochabstrakten und vereinfachten „Spielzeugproblemen“, auf echte Anwendungen außerhalb der KI konzentrierte.
Eine überraschend große Menge an Fachwissen wird gebraucht, um in einem sogar sehr begrenzten Aufgabenbereich ein „Experte“ sein zu können. Das spiegelt sich aber auch darin wieder, daß man als Mensch lange Zeit benötigt, um ein Experte zu werden. „Expertentum ist wissensintensiv“.(Man betrachte nur einmal das Schachspiel. Es wird vermutet, daß der Unterschied zwischen Anfänger und Könner fast ausschließlich in der Anzahl häufig erscheinender Muster liegt, mit denen man vertraut ist. Man schätzt, daß Profis zwischen 10000 und 100000 solcher Muster geläufig sind; das ist auch ungefähr die Größe des passiven Wortschatzes jemandes, der eine Sprache flüssig spricht.) In DENDRAL ist nicht nur Fachwissen über Methoden der Massenspektrometrie kodiert, sondern auch viel Information über Chemie und Graphentheorie – eine große Menge von Dingen, die man auch nur irgendwie zur Strukturbestimmung von Molekülen nutzen kann.
Für erfolgreiches „knowledge engineering“ muß man zunächst Entscheidungen über die Repräsentation des Wissens fällen, und zwar braucht man spezielle Darstellungen für spezielle Anwendungen. Eine

Darstellung, die für alle Aufgaben gilt, bringt nichts, wenn auch die Uniformität der Darstellung **innerhalb** eines Systems große Vorteile hat.
In den letzten Jahren gab es wieder vermehrte Anstrengungen im Bereich allgemeiner Architekturen für Intelligenz, die auf spezielles Fachwissen allgemeine Inferenzmethoden anwenden sollen. Keine solche Architektur könnte allerdings auf irgendeine offensichtliche oder direkte Weise lernen, sich wie der DENDRAL-Generator zu verhalten, dem zentralen Element des Systems.

Wieviel weiß DENDRAL?

Auf diese Frage gibt es keine schnelle Antwort, den Wissen ist immer mit Wissen über verwandte Dinge, mit dem Kontext, verwoben. Man könnte z. B. sagen, daß DENDRAL weiß, daß Kohlenstoff vierwertig ist (Kohlenstoffatome gehen vier Bindungen ein), seine Valenz ist 4. Es stellt sich die Frage, ob DENDRAL dann nicht auch wissen müßte, was Valenz ist, was ein Atom ist, und weitere verwandte Konzepte. Das Wissen, das DENDRAL über Valenz hat, ist ein völlig anderes als das eines Chemikers. Es wäre auch annähernd unmöglich, im Code von DENDRAL die Stellen zu nennen, an denen sich das Wissen über Valenz befindet, und wo es nicht ist.
Ein hilfreicher Ansatz zur Antwort dieser Frage ist die Unterscheidung zwischen „Wissen, wie“ und „Wissen, daß“. Doch auch dabei muß daran gedacht werden, daß verschiedene Programme, die das gleiche berechnen, ihr Wissen auf unterschiedliche Weise zwischen Fakten und Prozessen aufteilen können.
Jede Zusammenstellung darüber, was DENDRAL weiß (Konzepte, Fakten, Prozesse) ist also von begrenzter Beschreibungskraft. Hier trotzdem ein Versuch:

Das Wissen von DENDRAL über chemische Konzepte und Prozeduren

(Liste liegt bei)

Es gibt auch viele Dinge aus der Chemie im Zusammenhang mit dem Problem der Strukturbestimmung, die DENDRAL nicht weiß. Beispiele sind Polymere, dreidimensionale Geometrie jenseits von Stereoisomerie, viele chemische Eigenschaften wie Elektronegativität, Dipolmomente usw.
Weiterhin erwartet DENDRAL auch alles Wissen von LISP als Voraussetzung, z. B. arithmetische und symbolmanipulierende Operationen, etc.
Fast alles Wissen, das DENDRAL besitzt, ist maßgeschneidert auf das Problem der Strukturbestimmung von Molekülen. Obwohl manche Teile wie die Darstellung der Graphen und die Subgraphensuche breitere Anwendung gefunden haben, sind die problemlösenden Prozeduren in DENDRAL doch sehr komplex und speziell.

Beobachtungen über wissenschaftliche Entdeckungen

DENDRAL beweist, daß es möglich ist, die Interpretation von Daten und die Bildung von Hypothesen zu automatisieren. Außer der Aufgabe, mit dem sich dieses System beschäftigt, können, so glauben die Autoren von [1], viele andere Probleme von wissenschaftlichem Interesse auf die gleiche Weise automatisiert werden. Es werden folgende Verallgemeinerungen vorgeschlagen:

1– Hypothesenbildung in der Wissenschaft und die Lösung spezieller Aufgaben geschehen beide mit den gleichen Grundmethoden des Problemlösens.
   Meta-DENDRAL und Heuristic DENDRAL sind ähnlich organisiert. Beide arbeiten nach plan-generate-test, und beide sind stark nach dem Problem der Strukturbestimmung von Molekülen hin zugeschnitten. Obwohl Heuristic DENDRAL nach einzelnen Molekülen sucht und Meta-DENDRAL nach Regeln, war es ein überraschend kleiner Schritt vom ersteren zum letzteren.

2– Bei wissenschaftlichen Entdeckungen finden Ausschluß und Auswahl aus einer Menge möglicher Hypothesen statt, erhalten mit Hilfe heuristischer Untersuchungen. Dabei ist die Hypothesenmenge sehr mächtig; sie kann – wie bei der menschlichen Sprache – durch eine sehr große Kombinationsvielfalt einer endlichen Menge zustande kommen.
3– Beim Untersuchen der Menge der möglichen Hypothesen ist man stark von Anfangs-Annahmen beeinflußt.
Gilt das Constraint, daß eine bestimmte Unterstruktur nicht im Molekül vorhanden sein soll, dann werden Strukturen, die sie enthalten, eben nicht erzeugt. – Es sind nur diese Anfangsannahmen, die es erlauben, die Menge der zu untersuchenden Hypothesen überhaupt auf ein handhabbares Maß einzuschränken.
4– Wissenschaftliches Problemlösen und Entdecken beinhaltet auch die Anwendung einer großen Zahl vager und unbewiesener Ideen, mehr als die Anwendung von logischer Deduktion auf vorher bewiesene Sätze, Auch DENDRAL hat keine formale Theorie chemischer Stabilität als Basis, sondern benutzt viele schwache Teil-Annahmen.
In diesem Zusammenhang noch folgendes: Auch jedes experimentelle Datum könnte sich als falsch herausstellen. Deswegen kann es nicht einen strengen Test „Ist das die Lösung oder nicht?“ geben, sondern alle zur Verfügung stehenden Daten und Annahmen müssen gegeneinander aufgewogen werden.
5– Wissenschaftliches Problemlösen und Entdecken ist eine Mischung aus top-down- und bottom-up-Forschen, d. h. von der Erwartung bzw. von den empirischen Daten ausgehend. Beides ist notwendig. Heuristic DENDRAL ist hauptsächlich top-down angelegt, doch PLANNER, MOLION und PREDICTOR benutzen auch die empirischen Massenspektren.
6– Ein Generator, der nicht erschöpfend ist, ist nicht voll befriedigend, da man nicht mit Sicherheit sagen kann, welche Lösungen fehlen. (Er kann aber auch nützlich sein, wenn man nur irgendeine Lösung braucht.)
7– Ein Generator, der Lösungen u. U. mehrfach erzeugt, ist ebenfalls nicht voll befriedigend, da es sein kann, daß die Generierung nicht terminiert.
8– Wissen über Klassen von Hypothesen ist effizienter als Wissen über einzelne Hypothesen.
9– Eine kleine Menge von am Schluß übrigbleibenden Kandidaten ist genauso wertvoll wie ein einziger Kandidat.
Die obere Grenze der Anzahl von Kandidaten hängt aber vom Kontext ab.

Allgemein gesagt, geht es weniger darum, Regeln zu finden, die die besten aller möglichen Lösungskandidaten (bzw. wahre wissenschaftliche Sätze) produzieren, sondern darum, effiziente heuristische Regeln zu finden, um **vernünftige** Hypothesen zu finden.

Schlußbemerkungen

Obwohl es von Chemikern nur in begrenztem Maße gebraucht wurde, ist DENDRAL bei mit Computern arbeitenden Chemikern recht bekannt. Sie haben viele Teile des Systems in ihrer eigenen Software implementiert. DENDRAL läuft nun nicht mehr als einzelnes Softwarepaket; mangels einer genügend großen Benutzergemeinde gab es niemanden, der Zeit investiert hat, es zu pflegen.
Die größte Wirkung hatte DENDRAL auf die KI-Gemeinde, wo das Programm gut bekannt ist. Die Lehren, die von ihm kommen, sind nun Standard in der KI-Literatur und beim Entwurf und der Implementierung von Expertensystemen in der ganzen Welt.
Menschliches und künstliches Denken und Verstehen wird von verschiedenen Autoren auf drei verschiedene Weisen gesehen:
- Mind as Crystal, oder: Das Erkennen/Denken/Verstehen… sollte/könnte nach physikalischen Gesetzen gebildet werden.
- Mind as Chaos, oder: Denken/Erkennen… ist so komplex, daß man es so gut wie gar nicht enträtseln kann.

– Mind as Gadget, oder: KI ist eine Ingenieurwissenschaft. Ihre Aufgabe ist es, Systeme für spezielle, konkrete Zwecke zu entwerfen.

DENDRAL hat sicherlich den Geist des „Mind as Gadget“. Es kann weder Schachspielen, Kuchen backen, noch Krankheiten diagnostizieren, aber doch hat es eine gewisse Intelligenz, die durch allgemeine Strategien und spezielles Wissen gegeben ist und sich auch an neues Wissen anpassen kann. Diese allgemeinen Strategien der Programmentwicklung, kombiniert mit anderem Fachwissen, funktionieren auch für andere Aufgaben, das hat sich gezeigt.

LITERATUR:
[1] – Lindsay, R.K., B.G. Buchanan, E.A. Feigenbaum and J. Lederberg, DENDRAL: a case study of the first expert system for scientific hypothesis formation, Artificial Intelligence 61 (1993) 209-261. (Received April 1991, Revised June 1992)