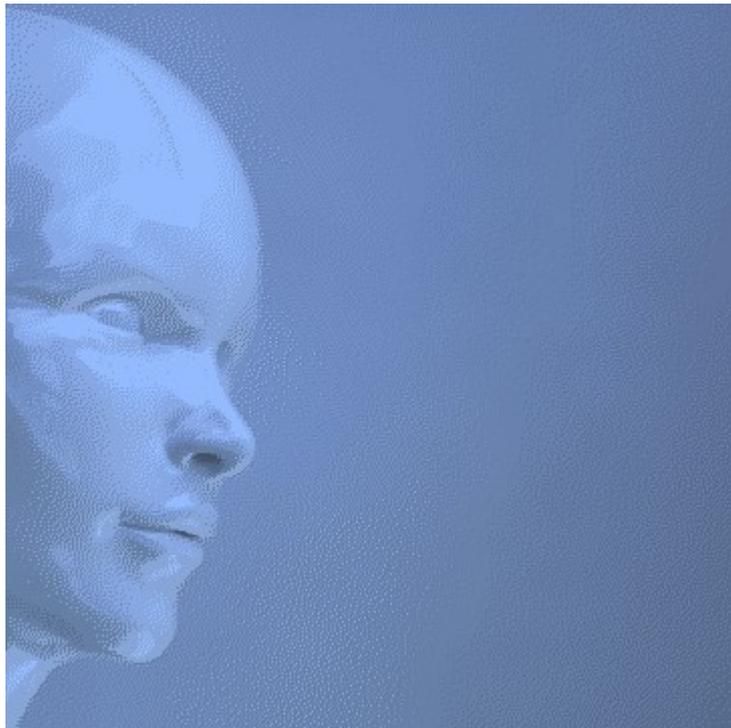


# Können Computer denken?

Entdecken Sie es selber



Diese Theorie ist nicht für das Jahr 2010 aktualisiert worden.

Wollen Sie helfen, zu übersetzen?

Bitte kontaktieren Sie mich per <http://mafait.org>.

## Inhaltsverzeichnis

1. Einleitung.....	3
1.1. Das Thinknowlogy Konzept.....	3
2. Die Algebra und Logik in der natürlichen Sprache.....	4
2.1. Eine Generalisierung / Spezifizierung-definition.....	4
2.1.1. Eine Beziehungsspezifikation.....	4
2.1.2. Verzamelingen (noch nicht übersetzt).....	5
2.1.2.1. Eine Spezifikationenmenge.....	5
2.1.2.2. Een relatie-verzameling (noch nicht übersetzt).....	5
2.1.2.3. Eine Generalisierungsmenge.....	5
2.1.3. Zuordnungen (nicht vollständig übersetzt).....	6
2.1.3.1. Eine Spezifizierungszuordnung (nicht vollständig übersetzt).....	6
2.1.3.2. Eine Generalisierungszuordnung.....	6
2.1.3.3. Zuordnungen und die grammatische Vergangenheitsform (n.n.ü.).....	7
2.2. Eine Auswahl.....	8
2.5. Selbständig Fragen stellen (nicht vollständig übersetzt).....	9
2.6. Das finden von grundsätzlichen Zusammenhängen.....	9
2.7. Semantic ambiguity.....	10
2.7.1. Autonomous semantic disambiguation.....	10
3. Das Aufbauen der Wissensstruktur.....	11
3.1. Der Bau einer Generalisierung / Spezifikation.....	11
3.1.1. Das Anlegen neuer Worte.....	12
3.1.2. Das Anlegen einer Spezifikationsverbindung.....	13
3.1.2.1. Das Anlegen einer Beziehungsverbindung (nicht vollständig übersetzt).....	13
3.1.3. Das Anlegen einer menge (nicht vollständig übersetzt).....	14
3.1.4. Das Anlegen einer Zuordnung (nicht vollständig übersetzt).....	14
3.1.4.1. Das Anlegen einer Spezifikationszuordnung (nicht voll. übersetzt).....	15
3.1.4.2. Das Anlegen einer Generalisierungszuordnung (nicht voll. übersetzt).....	15
3.2. Das Anlegen einer Auswahl.....	16
3.2.1. Die Ausführung von Selektionen.....	16
4. Dialoge.....	17
4.1. Die Grammatik.....	17
4.1.1. Das Lesen eines Satzes.....	17
4.1.1.1. Nur bekannte Strukturen werden akzeptiert.....	18
4.1.2. Antworten mit einem Satz.....	18
4.1.2.1. Selbständig Fragen stellen.....	18
5. Anwendungen.....	19
5.1. Programmieren in natürlicher Sprache.....	19
5.2. Das Beurteilen und Korrigieren von Texten und Spezifikationen.....	19
5.3. Das Automatisieren von Zusammenfassungen.....	20
5.4. Übersetzungsprogramme die den Kontext verstehen.....	20
5.5. Eine Suchmaschine die Fragen beantworten kann.....	20
5.6. Digitaler Dozent oder Selbststudium.....	21

## 1. Einleitung

- Kann man Computersoftware noch zu anderen Zwecken als zu veredeltem Rechnen, dem lagern von Daten und dem automatisieren von Betriebsprozessen gebrauchen?
- Werden wir jemals Fragen an Suchmaschinen stellen können anstelle von Suchen nach Wörtern oder Wortkombinationen?
- Werden Computer mit natürlicher Sprache umgehen können?
- Computer können Daten verarbeiten. Aber werden Computer jemals Information verarbeiten können? Oder anders gesagt: Werden wir jemals Information automatisieren.

Es sind viele Versuche unternommen worden um Computer 'denken' zu lassen. Wenn es aber kein Konzept bei diesen Versuchen gibt, werden diese Versuche fruchtlos bleiben. Nur mit einem gut durchdachten Konzept hat man eine Chance.

### 1.1. Das Thinknowlogy Konzept

Das Thinknowlogy Konzept hat die folgenden Prinzipien:

- 1) Programmiersprachen sind auf Algebra und Logik basiert.
- 2) Jeder Mensch hat ein angeborenes Gefühl für Algebra und Logik, auch wenn es bei dem einen ausgeprägter Entwickelt ist als bei dem anderen. In der natürlichen Sprache kommt dies zum Ausdruck.
- 3) Durch die Algebra und Logik in der natürlichen Sprache mit der Algebra und Logik aus den Programmiersprachen zu kombinieren ist es möglich um in natürlicher Sprache zu programmieren.
- 4) Das Endziel ist es um Information zu automatisieren.

Erst wird untersucht wie die Algebra und Logik in der natürlichen Sprache mit der Algebra und Logik aus den Programmiersprachen zu kombinieren ist so das es möglich ist um in natürlicher Sprache zu programmieren.

Später wird untersucht wie Informationen zu automatisieren sind.

## 2. Die Algebra und Logik in der natürlichen Sprache

Im Folgenden werden einige Aspekte der Algebra und Logik in der natürlichen Sprache beschrieben.

### 2.1. Eine Generalisierung / Spezifizierung-definition

Der Mensch organisiert seine Gedanken durch gruppieren, klassifizieren und zu generalisieren. Generalisieren heißt eigentlich die Hauptsachen von den Nebensachen zu trennen. Die Hauptsachen nennen wir **Generalisierungen** und die Nebensachen **Spezifizierungen**.

Der Unterscheidungsprozess von Haupt- und Nebensachen ist sogar im untersten Niveau der Sprache vorhanden. Man nehme den Satz: *“Johann ist ein Mensch.”*

*“Johann”* ist das Subjekt und *“Mensch”* ist eine Eigenschaft von *“Johann”*. Anders gesagt: *“Johann”* ist die Generalisierung von der Spezifikation *“Mensch”*, weil *“Mensch”* etwas spezifischer über das Subjekt *“Johann”* sagt.

Meer Spezifikationen über Johann:

- *“Johann ist ein Vater.”*
- *“Johann ist ein Bäcker.”*

In den vorhergehenden Sätzen wurde steht’s ein unbestimmter Artikel gebraucht: *“Johann ist ein Mensch.”*, *“Johann ist ein Vater.”* und *“Johann ist ein Bäcker.”*. Diese Struktur nennen wir im Thinknowlogy Konzept eine **Generalisierungs- / Spezifizierungsdefinition**.

#### 2.1.1. Eine Beziehungsspezifikation

Man nehme den Satz: *“Johann ist ein Freund von Mark.”*

Aus diesem Beispiel kann man ableiten das *“Johann”* eine *“Freund”*-Beziehung mit *“Mark”* hat. *“Freund”* wird hier die **Beziehungsspezifikation** genannt und *“Mark”* die Beziehung.

## 2.1.2. **Verzamelingen** (noch nicht übersetzt)

Hieronder wordt besproken hoe diverse woorden 'verzameld' worden.

### 2.1.2.1. **Eine Spezifikationenmenge**

Bei einer Generalisierung / Spezifizierung-definition kann eine Generalisierung auch mehrere Spezifikationen haben die zusammengehören. Zum Beispiel: “*Ein Glas ist voll oder leer.*”. Oder noch spezifizierter: “*Ein Glas ist voll, halb voll oder leer.*”.

So eine Gruppe von Spezifikationen nennen wir ein **Spezifikationenmenge**.

### 2.1.2.2. **Een relatie-verzameling** (noch nicht übersetzt)

Neem de zin: “*Jan is een vriend van Mark en Paula.*”.

In dit geval worden de relaties “*Mark*” en “*Paula*” verzameld, wat we dan ook een **relatie-verzameling** noemen.

### 2.1.2.3. **Eine Generalisierungsmenge**

Nehme die folgenden Sätze:

- “*Bush ist der ehemalige Präsident der Vereinigten Staaten von Amerika.*”
- “*Obama ist der derzeitige Präsident der Vereinigten Staaten von Amerika.*”

In diesen Fall sind die Spezifikation und die Beziehung gleich, aber die Generalisierung sind verschieden. Weil “*Bush*” und “*Obama*” hier scheinbar zusammen gehören müssen diese Generalisierungen versammelt werden.

Das nennen wir eine **Generalisierungsmenge**.

### 2.1.3. Zuordnungen (nicht vollständig übersetzt)

Eine **Zuordnung** ist eine Generalisierung / Spezifizierung-Struktur mit einem zugeordnetem Wert. Eine Generalisierung / Spezifizierung-Struktur ist statisch, während eine Zuordnungsstruktur dynamisch, abhängig vom Wert, ist.

Een toewijzing geeft dus de actuele toestand van een Generalisering / Spezifizierung aan en is te herkennen aan door het **bepaalde lidwoord** in die zin.

#### 2.1.3.1. Eine Spezifizierungszuordnung (nicht vollständig übersetzt)

Man nehme den Satz: “*Das Glas ist halb voll.*”.

Dies ist eine Generalisierung / Spezifizierung-Struktur, waarbij een bepaald lidwoord wordt gebruikt en de specificatie tot een Spezifikationenmenge behoort, bijvoorbeeld: “*Ein Glas ist voll, halb voll oder leer.*”. Das nennen wir eine **Spezifizierungszuordnung**.

Weil eine Zuordnung den aktuellen Zustand beschreibt, kann der Zustand sich auch ändern. Wenn das Glas ausgetrunken ist gilt “*Das Glas ist leer.*”.

Een voorbeeld van een Zuordnung van een Beziehungsspezifikation: “*Johann ist der Vater von Peter.*”.

#### 2.1.3.2. Eine Generalisierungszuordnung

Nehmen wir wieder die Sätze:

- “*Bush ist der ehemalige Präsident der Vereinigten Staaten von Amerika.*”
- “*Obama ist der derzeitige Präsident der Vereinigten Staaten von Amerika.*”

Dies sind auch Zuordnungen weil ein bestimmter Artikel gebraucht wird. Nun sind die Generalisierungen verschieden und formen eine Generalisierungsmenge. Deshalb werden diese Zuordnungen auch **Generalisierungszuordnungen** genannt.

### 2.1.3.3. Zuordnungen und die grammatische Vergangenheitsform (n.n.ü.)

Als een Zuordnung wijzigt, noemen we de oude toestand in de taalkunde: **verleden tijd**.

Voor het glas gold ooit: “Das Glas *ist* halb voll.”. Maar het glas is leeggedronken en nu geldt er:

- “Het glas *was* halfvol.”
- “Het glas *is* (nu) leeg.”

Ooit gold ook: “Bush *ist* der Präsident der Vereinigten Staaten von Amerika.”, maar de toestand is gewijzigd in:

- “Bush *ist* der *ehemalige* Präsident der Vereinigten Staaten von Amerika.” oder “Bush *war* der Präsident der Vereinigten Staaten von Amerika.”
- “Obama *ist* der (*derzeitige*) Präsident der Vereinigten Staaten von Amerika.”

## 2.2. Eine Auswahl

In der natürlichen Sprache gibt es auch Selektionen wie in diesem Satz: *“Wenn die Ampel gelb oder rot ist musst Du anhalten.”*. Das heißt wenn die Bedingung *“Die Ampel ist gelb oder rot”* wahr ist, gilt die Aktion *“musst du anhalten”*.

Oft gibt es auch Alternativen. Wenn die Ampel grün ist gilt: *“Du musst weiterfahren.”*. Wenn man dies zusammenfügt ergibt sich: *“Wenn die Ampel gelb oder rot ist musst du anhalten, sonst musst du weiterfahren.”*.

Eine Auswahl besteht also aus 2 oder 3 Teilen:

- Einer Bedingung (*“Wenn die Ampel gelb oder rot ist”*),
- einer Aktion (*“Du musst anhalten.”*)
- und eventuell einer alternativen Aktion (*“Du musst weiterfahren.”*).

Hier gilt die Generalisierung / Spezifizierung-Struktur:

- Die Bedingung: *“**Ein** Ampel ist gelb oder rot.”*;
- Die Aktionen: *“Du musst anhalten **oder** weiterfahren.”*.

Sowohl die Bedingung wie auch die Aktion sind auch Zuordnungen:

- Die Zuordnung einer Bedingung wird gebraucht um zu bestimmen ob die beschriebene Bedingung wahr ist. Die Bedingung aus dem Beispiel ist also wahr wenn eine der beiden Zuordnungen gilt: *“**Die** Ampel ist gelb.”* oder *“**Die** Ampel ist rot.”*;
- Abhängig von diesem Ergebnis wird dann die Zuordnung von der Aktion oder die alternative Aktion ausgeführt, also: *“Du musst anhalten.”* oder *“Du musst weiterfahren.”*.

Diese Theorie ist nicht für das Jahr 2010 aktualisiert worden.

Wollen Sie helfen, zu übersetzen?

Bitte kontaktieren Sie mich per <http://mafait.org>.

## ***2.5. Selbständig Fragen stellen (nicht vollständig übersetzt)***

Wenn dem System noch Information fehlen ist es möglich dass es automatisch Fragen an den Benutzer stellen kann. Auch gegensätzliche Information kann detektiert und durch Fragen korrigiert werden.

Hieronder wordt een methode uitgewerkt om het aanscherpen van informatie te stimuleren.

## ***2.6. Das finden von grundsätzlichen Zusammenhängen***

Es ist möglich um in einem Text automatisch nach grundsätzlichen Zusammenhängen zu suchen. Hierdurch kann das System auf Basis von schon bekannter Information seine Schlüsse ziehen. Dies wird später im Detail ausgearbeitet.

## 2.7. Semantic ambiguity

Two types of ambiguity can be distinguished: **static** ambiguity and **dynamic** (e.g. time-related) ambiguity.

An example of static ambiguity:

- “*Boston is a city in both the United States and the United Kingdom*”.

An example of dynamic ambiguity:

- “*Bush is inaugurated as (the) president of the United States.*”, because George H. W. Bush was inaugurated in 1989, and his son George W. Bush was inaugurated in 2001, and re-inaugurated in 2005.

### 2.7.1. Autonomous semantic disambiguation

When a sentence is entered and its context (semantics) is not clear, the system can either:

- use deduction to determine which context is meant by the user;
- make an assumption, when the meant context cannot be determined, but when it is quite obvious;
- or ask a question, when the system has no clue about the context.

### 3. Das Aufbauen der Wissensstruktur

Jemand mit Programmiererfahrung hat im vorherigen Kapitel wahrscheinlich einige Grundprinzipien der Programmierung entdeckt. Ein **Assignment** in der Form einer Zuordnung, eine **if-then-else** Struktur in der Form einer Auswahl und vielleicht auch eine **Variabelendeklaration** in der Form einer Generalisierung / Spezifizierung-definition: Man gibt einer Variablen Struktur, kennt aber noch keinen Wert zu.

Mit dieser Übereinstimmung zwischen natürlicher Sprache und Programmiersprachen kann eine Wissensstruktur aufgebaut werden. Hierdurch entsteht eine Verbindung von natürlicher Sprache und Programmiersprache und ist es in Prinzip möglich um in einer natürlichen Sprache zu programmieren. Dies kann die Basis eines 'denkenden' Computers werden.

Im Folgenden wird per Element und mit Sätzen aus der natürlichen Sprache erklärt wie eine Wissensstruktur aufgebaut werden kann.

#### 3.1. Der Bau einer Generalisierung / Spezifikation

Ein wichtiges Basiselement der Wissensstruktur ist die Generalisierung / Spezifizierung. Nachstehend werden die verschiedenen Aspekte dieser Struktur beschrieben und wie man damit eine Wissensstruktur aufbaut.

### 3.1.1. Das Anlegen neuer Worte

Man nehme den Satz: “*Johann ist ein Vater.*”.

Das System muss erst die Worte “*Johann*” und “*Vater*” kennen bevor die Struktur gebaut werden kann. Wenn es diese Wörter noch nicht im System gibt oder von einer undefinierten Wortart sind, müssen diese Wörter erst angelegt werden.

#### **Johann**

Das Wort “*Johann*” fängt mit einem Großbuchstaben an und ist das erste Wort des Satzes. Es kann also ein Eigename oder ein Substantiv sein. Es kann auch ein unbekannter Worttyp sein wenn es nicht mit einem Großbuchstaben beginnt.

In diesen Fall werden drei Wörter angelegt, nämlich:

- “*Johann*” vom Typ *Eigennamen*;
- “*Johann*” vom Typ *Nomen*;
- “*johann*” (ohne Kapital) als undefinierter Worttyp.

Während des Aufbaus der Wissensstruktur wird nur ein der drei Wörter genutzt. Dann werden die nicht benutzten Wörter entfernt damit das System sauber bleibt. Hiervon später mehr.

#### **Vater**

Das Wort “*Vater*” wird hat einen vorgestellten Artikel, es ist darum eindeutig ein Substantiv.

In dem Satz “*Ein Glas ist voll, halb voll oder leer.*” ist es nicht eindeutig von welchem Type die Wörter “*voll*”, “*halb voll*” und “*leer*” sind. Es wird aber angenommen das es sich um denselben Worttyp handelt.

### 3.1.2. Das Anlegen einer Spezifikationsverbindung

Jetzt wo das System die Wörter kennt, können die Wörter miteinander verbunden werden. Besser gesagt: Die Generalisierung wird mit der Spezifikation verbunden.

Ein Beispiel: “*Johann ist ein Vater.*” und “*Johann ist ein Bäcker.*”.

Beim ersten Satz wird vom Generalisierungswort “*Johann*” eine Spezifikationsverbindung zur Spezifizierung “*Vater*” gelegt und im zweiten Satz von “*Johann*” zu “*Bäcker*”.

In der Wissensstruktur ist jetzt deutlich dass Johann sowohl Vater als auch Bäcker ist.

#### 3.1.2.1. Das Anlegen einer Beziehungsverbindung (nicht vollständig übersetzt)

Um eine Beziehungsspezifikation “*Johann ist ein Freund von Mark.*” an eine Wissensstruktur zu zufügen wird das Generalisierungswort “*Johann*” durch eine Spezifikationsverbindung verbunden mit der Spezifikation “*Freund*” und mit der Beziehung “*Mark*”.

Es wird also eine Spezifikationsverbindung mit zwei Verweisungen benutzt. Diese spezielle Spezifikationsverbindung wird auch **Beziehungsspezifikationsverbindung** oder **Beziehungsverbindung** genannt.

Die Beziehungsverbindung wird wie folgt gelesen: “*Johann*” hat eine “*Freund*”-Beziehung mit “*Mark*”. Das gleiche gilt für “*Bush*” und “*Obama*” die eine “*Präsident*”-Beziehung mit den “*Vereinigten Staaten von Amerika*” haben.

Bij de zin “*Johann ist ein Freund von Mark und Paula.*” worden twee relatie-verbindingen gecreëerd vanaf “*Jan*”: Beide met specificatie “*vriend*”, maar één met een relatie-verwijzing naar “*Mark*” en één met “*Paula.*”.

### 3.1.3. Das Anlegen einer Menge (nicht vollständig übersetzt)

Nimm den Satz: *“Ein Glas ist voll, halb voll oder leer.”*.

De **specificatie-woorden** van generalisatie *“Glas”* worden ‘verzameld’ door ze wederzijds met verzamel-verbindingen aan elkaar te verbinden:

- *“voll”* wird mit einer auflaufenden Mengenbeziehung an *“halb voll”* gekoppelt;
- *“halb voll”* wird mit einer auflaufenden Mengenbeziehung an *“leer”* gekoppelt;
- *“leer”* wird mit einer ablaufenden Mengenbeziehung an *“halb voll”* gekoppelt;
- *“halb voll”* wird mit einer ablaufenden Mengenbeziehung an *“voll”* gekoppelt.

Das ‘ablaufen’ oder ‘auflaufen’ von einer Mengenbeziehung ist von der Reihenfolge in den Aufzählungen abhängig.

Het verzamelen van relaties en generalisaties gebeurt op dezelfde manier:

#### Het verzamelen van generalisatie-woorden

- *“Bush ist der ehemalige Präsident der Vereinigten Staaten von Amerika.”*
- *“Obama ist der derzeitige Präsident der Vereinigten Staaten von Amerika.”*

#### Het verzamelen van relatie-woorden

- *“Jan is de vader van Piet en Marie.”*

### 3.1.4. Das Anlegen einer Zuordnung (nicht vollständig übersetzt)

Eine Zuordnung ist eine Generalisierung / Spezifizierung-Struktur mit einem zugeordnetem Wert. Deshalb wird erst eine Generalisierungs- / Spezifikationsstruktur angelegt, **als diese noch nicht besteht**, und dann eine Zuordnungsstruktur.

### 3.1.4.1. Das Anlegen einer Spezifikationszuordnung (nicht voll. übersetzt)

Een Spezifizierungszuordnung met een Spezifikationenmenge kan maar één van de toestanden tegelijk weergeven, als het voegwoord “oder” in de zin wordt gebruikt.

Wir nehmen wieder den Satz “*Ein Glas ist voll, halb voll oder leer.*” und den Zustand (Zuordnung) “*Das Glas ist halb voll.*”.

Wenn das Glas jetzt ausgetrunken wird gilt der Zustand: “*Das Glas ist leer.*”. Die Zuordnung von “*Glas*” zu “*halb voll*” wird deaktiviert, so dass diese Zuordnung wie die grammatische Vergangenheitsform interpretiert wird. Dann wird eine neue Zuordnung von “*Glas*” zu “*leer*” jetzt angelegt um den heutigen Zustand von “*Glas*” fest zu legen.

### Mehrere Zuordnungen pro Generalisierung

Een Spezifizierungszuordnung kan meerdere toewijzingen hebben. Zo kunnen beide toewijzingen tegelijk gelden: “*Das Glas ist leer.*” und “*Das Glas ist blau .*”.

### Een toewijzing met meerdere relaties

Bij het creëren van een Spezifizierungszuordnung met meerdere relaties, wordt voor elke relatie een aparte specificatie-toewijzing gecreëerd.

Twee opmerkingen:

- In de zin “*Jan is de vader van Piet en Marie.*” geeft het voegwoord “*en*” aan, dat beide relaties aan elkaar zijn verbonden en dus tegelijk gelden;
- Ook als je daarna zegt: “*Jan is de bakker van Paul.*”, blijven de bovenstaande toewijzingen actief, omdat “*vader*” en “*bakker*” deze geen verzameling vormen.

### 3.1.4.2. Das Anlegen einer Generalisierungszuordnung (nicht voll. übersetzt)

Ook Generalisierungszuordnungen kunnen maar één toestand tegelijk weergeven, omdat ze door een Generalisierungsmenge aan elkaar zijn verbonden:

Wenn erst Zuordnung gilt “*Bush ist der Präsident der Vereinigten Staaten von Amerika.*” und nach den Wahlen gilt “*Obama ist der Präsident der Vereinigten Staaten von Amerika.*”, dann muss die Zuordnung von “*Bush*” zu “*der Präsident der Vereinigten Staaten von Amerika*” deaktiviert werden, so dass diese Zuordnung wie die grammatische Vergangenheitsform interpretiert wird. Dann muss eine neue Zuordnung von “*Obama*” zu “*der Präsident der Vereinigten Staaten von Amerika*” angelegt werden.

### 3.2. Das Anlegen einer Auswahl

Genau wie die Generalisierung / Spezifizierung ist die Auswahl auch ein wichtiges Basiselement der Wissensstruktur.

Um eine Wissensstruktur aus einer Auswahl auf zu bauen werden erst alle Generalisierung / Spezifizierung-strukturen aus denen die Auswahl besteht angelegt. Dann werden die Bedingungen in einer [Bedingungsliste](#) bewahrt, die Aktionen in einer [Aktionsliste](#) und die Alternativen Aktionen in einer [Alternative-Aktionsliste](#).

Nimm den Satz: *“Wenn die Ampel gelb oder rot ist, musst Du stoppen sonst kannst Du weiterfahren.”*.

Erst wird also die Generalisierung / Spezifizierung *“Die Ampel ist gelb oder rot.”*, *“Du musst stoppen.”* und *“Du kannst weiterfahren.”* angelegt. Dann wird *“Die Ampel ist gelb oder rot.”* in die Liste der Bedingungen gesetzt mit einer Verweisung zu der zugehörigen Aktion und Alternativen Aktion. *“Du musst stoppen.”* wird in die Aktionsliste gesetzt und *“Du kannst weiterfahren.”* kommt in die Alternative-Aktionsliste.

#### 3.2.1. Die Ausführung von Selektionen

Nachdem das System den Satz gelesen und verarbeitet hat werden alle Bedingungen in der Wissensstruktur abgearbeitet und die zugehörigen Aktionen oder Alternativen Aktion ausgeführt. Dieser Prozess wird wiederholt bis es keine Veränderungen mehr im System gibt. Das System ist dann 'in Ruhe' und kann jetzt den folgenden Satz lesen und verarbeiten.

Hierdurch sieht es so aus als ob das System 'Denkt', aber eigentlich wird einfach nur ein Programm abgearbeitet das in natürlicher Sprache geschrieben ist.

## 4. Dialoge

In den vorigen Beispielen waren die Generalisierungen gleich und die Spezifikationen verschieden. Wir haben Sätze in natürlicher Sprache in eine Wissensstruktur übersetzt. Um aber einen Dialog mit dem Benutzer führen zu können muss das System auch fähig sein zu antworten. Hierfür wird die Wissensstruktur wieder in natürliche Sprache übersetzt.

### 4.1. Die Grammatik

Für das Lesen und das Antworten in einer bestimmten Sprache ist es nötig das daß System die Grammatik dieser Sprache kennt. Für jede gewünschte Sprache muss also zuvor die Grammatik definiert werden.

Um das System flexibel zu gestalten wird diese Grammatik nicht 'hard-coded' programmiert sondern als Textdatei gesichert. Beim Programmstart werden dann ein oder mehrere Grammatik-Dateien eingelesen.

Es ist möglich um Dialoge in mehreren Sprachen zu führen und schließlich müsste es auch möglich sein um das System automatisch übersetzen zu lassen, aber davon später mehr.

#### 4.1.1. Das Lesen eines Satzes

Die Grammatik wird beim lesen eines Satzes benutzt:

- Um zu kontrollieren ob der Satz grammatikalisch korrekt ist;
- Um zwischen verschiedenen Strukturen wie Generalisierung / Spezifizierung, Zuordnung und Auswahl zu unterscheiden;
- Um bestimmte Wörter vorab zu definieren, aber davon später mehr.

#### **4.1.1.1. Nur bekannte Strukturen werden akzeptiert**

In den vorigen Kapiteln haben wir gesehen das für das Umsetzen von Sätzen in eine Wissensstruktur pro Satzaufbau (Generalisierung / Spezifizierung, Zuordnung und Auswahl) ein Konzept nötig ist um die Information aus den Sätzen in die Wissensstruktur zu übersetzen.

Oder anders gefolgert: Das System kann nur Satzstrukturen akzeptieren die in der Grammatikdefinition festgelegt sind, nur für diese Strukturen gibt es ein Konzept um die Satzinformation auch in die Wissensstruktur um zu setzen. Dies ist eine Einschränkung des Systems, denn eigentlich wollen wir alle angebotenen Sätze lesen und die darin enthaltene Information in die Wissensstruktur aufnehmen.

**Das Ziel dieses Projekts ist dann auch um für soviel möglich Satzstrukturen ein Konzept zu bedenken daß die Information aus dem Satz extrahiert und in die Wissensstruktur umgesetzt.**

Es wird noch viel Zeit und Mühe kosten um dieses Ziel zu erreichen. Dann wird es aber möglich sein um Information automatisch zu verarbeiten und dann kommt der 'denkende' Computer in greifbare Nähe.

#### **4.1.2. Antworten mit einem Satz**

Die Grammatik wird gebraucht um die Wörter die das System dem Benutzer präsentieren will in einen grammatikalisch korrekten Satz zu fügen.

##### **4.1.2.1. Selbstständig Fragen stellen**

Das System kann Lücken und Gegensätze in der Wissensstruktur erkennen, aber dies wird später noch erläutert.

Außerdem kann das System auch selbständig Fragen an den Benutzer stellen um die gefundenen Lücken und Gegensätze zu korrigieren. Das ist intelligentem Handeln ähnlich und verstärkt das Gefühl das Computer denken können.

## 5. Anwendungen

Wenn die vorgestellte Theorie genug weiterentwickelt wird, können Computer schriftliche Informationen, wie zum Beispiel Texte aus dem Internet, Dokumente von Betrieben, Schulbücher und technische Entwürfe von Software Entwicklern, lesen und 'begreifen'.

Nachfolgend eine Reihe von Möglichkeiten um das System ein zu setzen wenn die Grammatik ausreichend Satzstrukturen beherrscht.

### 5.1. Programmieren in natürlicher Sprache

*Durch das Auswahlkonzept ist es möglich um dem System automatisch logische Aufträge in natürlicher Sprache zu geben. Hierdurch kann man in natürlicher Sprache programmieren. So wäre es zum Beispiel möglich um die Spielregeln eines Spieles einzulesen und dann gegen den Computer zu spielen ohne das Spiel vorher zu programmieren.*

#### **Softwareentwicklung**

*Ein ernsthafteres Beispiel wäre es um einen technischen Entwurf aus der Software-Entwicklung ein zu lesen und dann bis zu einem bestimmten Niveau auszuführen ohne zu programmieren. Es sollte minimal als Prototype System dienen, aber in Prinzip kann hiermit die Software-Entwicklung automatisiert werden.*

*Es ist also eigentlich das automatisieren der Automatisierung. Ist das Kannibalismus oder unvermeidlicher Fortschritt?*

### 5.2. Das Beurteilen und Korrigieren von Texten und Spezifikationen

Das System kann Lücken und Gegensätze in Texten finden, auch Lücken und Gegensätze an die ein Mensch nicht direkt denken würde. Das System kann sie anweisen und mit der Hilfe von spezifischen Fragen eventuell auch reparieren. Auf Dauer sollte es möglich sein daß das System Texte und Spezifikationen beurteilt und korrigiert.

### **5.3. Das Automatisieren von Zusammenfassungen**

Das Zusammenfassen von Texten ist prinzipiell nichts anderes als das Trennen der Nebensachen von den Hauptsachen, wobei solange Nebensachen weggelassen werden bis der Text die gewünschte Länge hat.

Wenn Zusammenfassungen von Dokumenten automatisch generiert werden können ist das sehr praktisch für Manager die oft mit dicken Dossiers in eine Besprechung gehen. Sie können dann auf einfache Weise eine Zusammenfassung in der gewünschten Länge machen um doch noch ausreichend über den Inhalt informiert zu sein.

### **5.4. Übersetzungsprogramme die den Kontext verstehen**

Die meisten Übersetzungsprogramme übersetzen Wort für Wort und erkennen keine Ausdrücke. Weil Wörter oft mehrere Bedeutungen haben und die Übersetzungsprogramme den Kontext davon nicht verstehen, ist die automatische Übersetzung oft von schlechter Qualität.

Dieses System kann prinzipiell den Kontext 'verstehen', hierdurch müsste es möglich sein die Qualität der automatischen Übersetzungen auf ein höheres Niveau zu heben.

### **5.5. Eine Suchmaschine die Fragen beantworten kann**

Zur Zeit wählen wir noch selbst die Stichwörter um das Internet zu durchsuchen und hoffen das sie relevante Webseiten finden werden die unsere Fragen beantworten.

Auf diese Art ist es recht einfach um Ausdrücke und unbekannte Wörter zu finden. Es wird schwieriger wenn es um eine Frage geht zu deren Beantwortung Sachkenntnis nötig ist, wo der Ausdruck unbekannt ist oder der genaue Kontext nicht bekannt ist.

Und in welcher Sprache probiert man die Information zu suchen? Wahrscheinlich wird man englische Wörter benutzen. Aber was soll man machen wenn man die korrekte Übersetzung nicht kennt oder die Information nur in chinesisch vorhanden ist? Wie bekommt man dann Antwort auf seine Frage, **als je die taal niet beheerst?**

Man kann nur dann Fragen an eine Suchmaschine stellen wenn diese Maschine die vorhandene Information 'begreift' und im richtigen Kontext sieht.

## **5.6. Digitaler Dozent oder Selbststudium**

Ein Dozent bestimmt am Anfang des Schuljahres an der Hand der vorhandenen Schulbücher und der Lehrstunden, welche Kapitel und Themas in dem Jahr behandelt werden. Vor jeder Stunde beschreibt der Lehrer seinen Schülern den zu behandelnden Stoff. Außerdem kann der Lehrer Fragen über den /Stoff beantworten weil er den Stoff bis ins Detail beherrscht.

Wenn es gelingt um automatisch Zusammenfassungen zu machen (vom zum behandeln Stoff) und wenn es gelingt um Suchmaschinen Fragen beantworten zu lassen, dann müsste es auch möglich sein das lehren eines theoretischen Fachs zu automatisieren oder um auf jeden Fall um einen guten digitalen Lehrer als Selbststudienhilfe anzubieten.