

---

# Data Warehousing und Data Mining


## Eine Einführung in entscheidungsunterstützende Systeme

Interaktive Folien zu Kapitel 4  
*Regelbasierte Systeme*

*“Kasparow hat  
mit Deep Blue  
gleichgezogen”*

# Einordnung

## Nutzwertanalyse am Beispiel von AHP

Kioskstandort , Personalauswahl

## Was-Wenn-Analyse

Erfolgsrechnung 



Anzeigenplanung , Produktionsplanung

## ⇒ **Regelbasierte Systeme**

⇒ **Spesen** , **Betriebskredit**

⇒ **Regelverkettung**

## Data Warehouses

- Anlageberatung 
- Lieferfrist, Handel, Verkauf 


## Data Mining - Ein Überblick

- Zeitschriften , Bank

## Regelinduktion

- Spesen , Bonitätsklassifikation 

## Neuronale Netze

- Bonitätsklassifikation , Bonitätsvorhersage
- EindimPerzeptron, ZweidimPerzeptron
- MehrklassPerzeptron
- MehrstufPerzeptron

---

# Unterrichtsmaterial

---

## Software

- Demonstrationsversion von *XpertRule KBS*  
(Regelbasierte Systeme aus Entscheidungsbäumen)
- *Visual Basic* für Applikationen unter *MS Excel*

## Beispiele und Aufgaben

- *Spesen* mit *XpertRule* 📌
- *Betriebskredit* mit *XpertRule*
- *Regelverkettung* mit *VBA*

## Produktinformation

- <http://www.attar.com/>

---

# Grundlagen

---

## Grundlagen

⇒ Beispiele <i>Künstlicher Intelligenz</i>	<u>6</u>
⇒ Schwierigkeiten	<u>8</u>
⇒ Expertensysteme	<u>10</u>
⇒ Wissen als Regeln darstellen	<u>21</u>

## Entwicklung mit *XpertRule*

• Spesen 	<u>31</u>
• Betriebskredit	<u>45</u>
• Wissen erklären	<u>49</u>

## Theorie

• Einfache Regeln darstellen	<u>56</u>
• Schlüsse aus einfachen Regeln ableiten	<u>59</u>
• Regelverkettung mit <i>VBA</i>	<u>62</u>

# Beispiele der Künstlichen Intelligenz

---

Sprachverarbeitung

Maschinelles Lernen

Bildverarbeitung

...

Anwendung und Weiterentwicklung in  
**Expertensystemen** und **Data Mining**

Expertensysteme sind meist **regelbasiert**

# Einige Anwendungen der Künstlichen Intelligenz

Anwendung	Eignung
Erkennen isolierter Sprache (Bsp. <i>CAD, Roboter</i> )	relativ gut
Verstehen geschriebener Einfachsprache (Bsp. Übersetzung eines <i>Wetterberichts</i> )	brauchbar
Steuerung nicht autonomer Roboter	gut
▸ Data Mining →	relativ gut
Problemlösung durch Expertensysteme →	relativ gut

# Löst die KI praktische Probleme?

Ungelöst, aber “leicht”

⇒ Handschriften lesen

⇒ Kontinuierliche Sprache verstehen

⇒ Sprache übersetzen

Teils gelöst, obwohl “schwierig”

Gegen Menschen spielen

(*Schach*)

Mathematische Ausdrücke manipulieren

(*symbolisch differenzieren*)

Fachexperten in engen Bereichen ersetzen

(*Hardware konfigurieren*)

Muster in Massendaten entdecken

(*Bonitätsbeurteilung*)



# Ungelöste Probleme an der Sprachübersetzung

## Gesprochenes übersetzen

*“Give me a new display”*

*“Give me a nudist play”*

## Schriftliches übersetzen

*“Time flies like an arrow”*

### **syntaktisch mehrdeutig**

*time*, *fly* = Substantive oder Verben?

*like* = Konjunktion oder Verb?

### **semantisch mehrdeutig**

*like* = “wie” oder *like* = “mögen”

Beschränkung auf einfache Gebrauchstexte!

---

## 📌 Reparaturassistent - Ein Expertensystem

---

<i>Entwickler</i>	<i>Campbell's Soup Company</i>
<i>Ziel</i>	Reparatur von Sterilisiergeräten
<i>Anlass</i>	Pensionierung eines Technikers
<i>Alternativen</i>	Neuer Techniker Schulung Beratungssystem
<i>Werkzeuge</i>	Personalcomputer Expertensystem-Shell

# Begriff des Expertensystems

---

Expertensystem :=

Applikation, die Methoden der **Künstlichen Intelligenz** anwendet, indem sie ...

Wissen auf einem **eng begrenzten Gebiet** ...

**problemangepasst**, **änderungsfreundlich** und **verarbeitungseffizient** darstellt, ...

›algorithmisch oder ›heuristisch **Schlüsse** daraus zieht und ...

diese Schlüsse unter Bezug auf Falldaten im Dialog mit dem Benutzer **erklärt**.

---

# Expertensysteme sind meist regelbasiert:

---

**WENN**

Funktion = "Vizedirektor" und  
Hotelklasse = "Komfort"

**DANN**

Spesenantrag ablehnen

*Verallgemeinerung*

**WENN**

Bedingung

**DANN**

Folgerung

# Erklärungen in regelbasierten Systemen



**Hat Meier Anspruch auf finanzielle Unterstützung ?**

Wie alt ist Meier? 23

Hat Meier eine Niederlassungsbewilligung? ja

...

Wie gross sind die Ersparnisse von Meier? 2600

**Meier hat keinen Anspruch auf Unterstützung. Weshalb ?**

Die folgende Bestimmung ist nicht erfüllt:

“Ein Antragssteller hat Anspruch auf Hilfe, falls er

(1) volljährig ist

(2) niederlassungsberechtigt ist

...

(5) finanzielle Hilfe benötigt”

Die Voraussetzung (5) trifft nicht zu. Weshalb ?

...

## 4.1 Daten - und Wissensverarbeitung

	<b>Datenverarbeitung</b>	<b>Wissensverarbeitung</b>
<i>Gegenstand</i>	einfache Massendaten	komplexes Wissen über einen engen Bereich
<i>Wertebereich</i>	meist numerisch	meist symbolisch
<i>Verarbeitung</i>	meist numerisch	herleitend
<i>Probleme</i>	genau definierbar	schlecht definierbar
<i>Methoden</i>	algorithmisch	auch heuristisch
<i>Entwicklung</i>	planbare Phasen	explorativ
<i>Erklärung</i>	Bedienungsanleitung	Herleitungserklärung

### Begriffe

	<i>Definition</i>	<i>Beispiel</i>
<b>algorithmisch</b>	Die Wissensbasis führt zu einer Maximallösung	Lineare Optimierung
<b>heuristisch</b>	Die Wissensbasis verbessert naive Lösungen inkrementell	Monte Carlo-Simulation
<b>explorativ</b>	ungeplant entdeckend	

---

# Numerische und symbolische Verarbeitung

---

## ① Numerische Verarbeitung

Manipulation von *Zahlen*

Nutzwertanalyse am Beispiel von AHP

Was Wenn-Analyse

- Neuronale Netze

## ② Symbolische Verarbeitung

Manipulation von *Zeichenketten*

⇒ Regelbasierte Systeme →

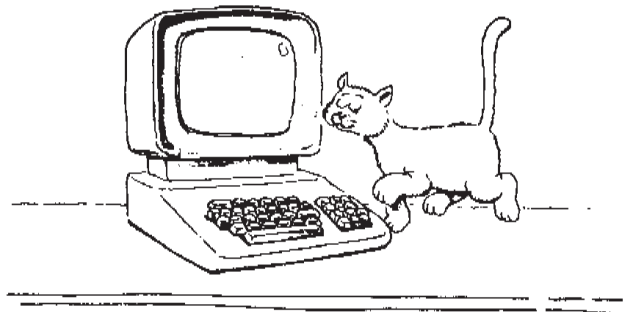
- Information Retrieval
- Symbolisches Rechnen  
(z.B. Differenzieren mit der Kettenregel)

# Expertensysteme - Eine Definition

Expertensysteme ...

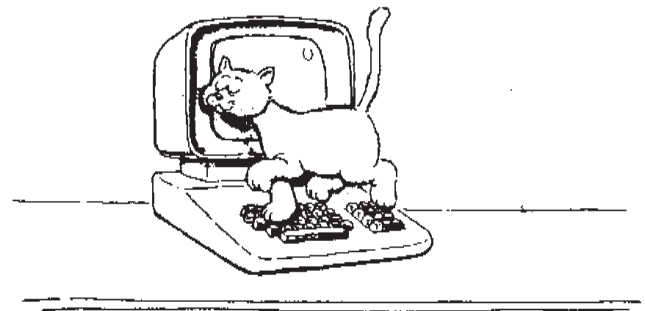
... stellen *Wissen* dar

- auf eng begrenzten Gebieten
- problemangepasst
- änderungsfreundlich
- verarbeitungseffizient



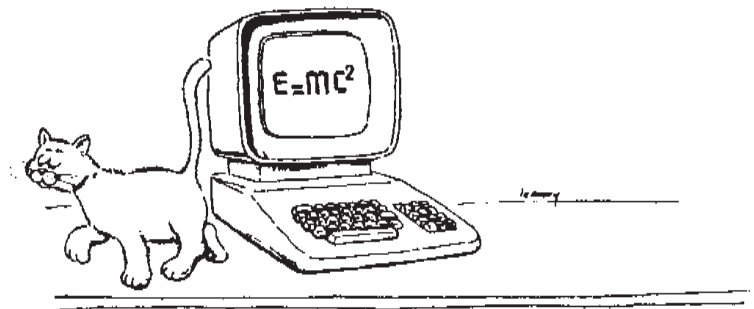
... ziehen *daraus Schlüsse*

- algorithmisch
- heuristisch



... *erklären* diese Schlüsse

- unter Bezug auf Falldaten
- im Dialog mit dem Benutzer.





## Leasingberater

Finanzierungskosten?

Abschlussrechnung?

Risiken?

### Leasing/Kreditkauf-Entscheidung - Entscheidungsmodell



Phase	Inhalt
<b>Zielbildung</b>	Finanzierungskosten, Liquidität, Bilanzwirkungen, Risiken (Untergang, Verwertung, ...), Flexibilität
<b>Datenerfassung</b>	Leasingvertrag, Kreditvertrag, Investitionsgut, Unternehmung
<b>Analyse</b>	Einzahlungen/Auszahlungen (insb. Annuitäten), Bilanz, ▸ Erfolgsrechnung, ▸ Kapitalflussrechnung, Kennzahlen
<b>Bewertung und Entscheidung</b>	Nutzwertanalyse zur Gewichtung der Knoten einer Zielhierarchie ( <u>AHP</u> )

## 4.5 Architektur



### Beispiele

- Wissensbasis: *Regeln der Stipendienvergabe*
- Wissenserwerb: *Änderungen des Stipendienreglements*
- Erklärung: *Warum hat X Anspruch auf finanzielle Unterstützung?*

## 4.7 Wissenskassen

<i>Wissen</i>	<i>Definition</i>	<i>Methodenbeispiel(e)</i>
<b><i>symbolisch</i></b>	nichtnumerisches Wissen darstellen / herleiten	Regelbasierte Systeme
<b><i>numerisch</i></b>	numerisches Wissen darstellen und herleiten	▸ Neuronale Netze
<b><i>deduktiv</i></b>	Besonderes aus Allgemeinem ableiten	Regelbasierte Systeme
<b><i>induktiv</i></b>	Allgemeines aus Besonderem ableiten	▸ Neuronale Netze, ▸ Inferenzstatistik
<b><i>deterministisch</i></b>	Sicheres Wissen darstellen und herleiten	Lineare ▸ Optimierung
<b><i>probabilistisch</i></b>	Unsicheres Wissen darstellen und herleiten	▸ Inferenzstatistik, Fuzzy Logik
<b><i>deklarativ</i></b>	"Was" spezifizieren	Regelbasierte Systeme
<b><i>prozedural</i></b>	"Wie" spezifizieren	Programm in ▸ VBA
<b><i>algorithmisch</i></b>	Die Wissensbasis führt zu einer Maximallösung	Lineare ▸ Optimierung
<b><i>heuristisch</i></b>	Die Wissensbasis verbessert naive Lösungen	Regelbasierte Systeme, ▸ Monte Carlo-Simulation

# Expertenwissen darstellen und herleiten

Expertensysteme arbeiten meist ...

symbolisch

deduktiv

deklarativ

heuristisch

Wissen oft in Form von Regeln

Regelbasierte Systeme

# Wissen als Regeln darstellen

WENN **der Hahn kräht auf dem Mist**,  
DANN ändert sich das Wetter  
ODER es bleibt wie es ist

WENN **Bedingung**  
DANN **Folgerung**

## Einfache Folgerung

WENN Bedingung DANN **Folgerung**

## Zusammengesetzte Folgerung

WENN Bedingung DANN **Folgerung1** UND/ODER **Folgerung2**

# Regel aus Bedingung und Folgerung

WENN **Bedingung**  
DANN **Folgerung**

## Bedingung

- **Attribut Wert** [ UND | ODER **Attribut Wert** ... ]  
"WENN der Hahn *kräht auf dem Mist*"

## Folgerung

- **Attribut Wert** [ UND | ODER **Attribut Wert** ... ]  
"DANN *ändert sich* das Wetter ODER es *bleibt wie es ist*"
- Statt eines Attribut/Wert-Paars kann auch eine Aktion stehen (Anweisung, die weder wahr noch falsch ist)  
"Gehe trotzdem skilaufen"

---

# Attribut, Wert und Datentyp

---

Werkzeuge zur Entwicklung von Expertensystemen unterscheiden ...

- **Attribut**

*Wetter*

- **Wert**

*veränderlich*

- **Datentyp** →

*Aufzählungstyp [gut, veränderlich, schlecht]*

# Datentyp

**Datentyp** := ( Wertebereich, Operationen )

**Zahl** (engl. numeric)

*Mengenumsatz*

- Wertebereich: positive Ganzzahlen
- Operationen: arithmetische Grundoperationen

**Aufzählung** (engl. enumeration, list)

*Zivilstand*

- Wertebereich { ledig, verheiratet, verwittwet, geschieden }
- Beispieloperation: Element von

**Wahrheitswert** (engl. boolean)

*verheiratet*

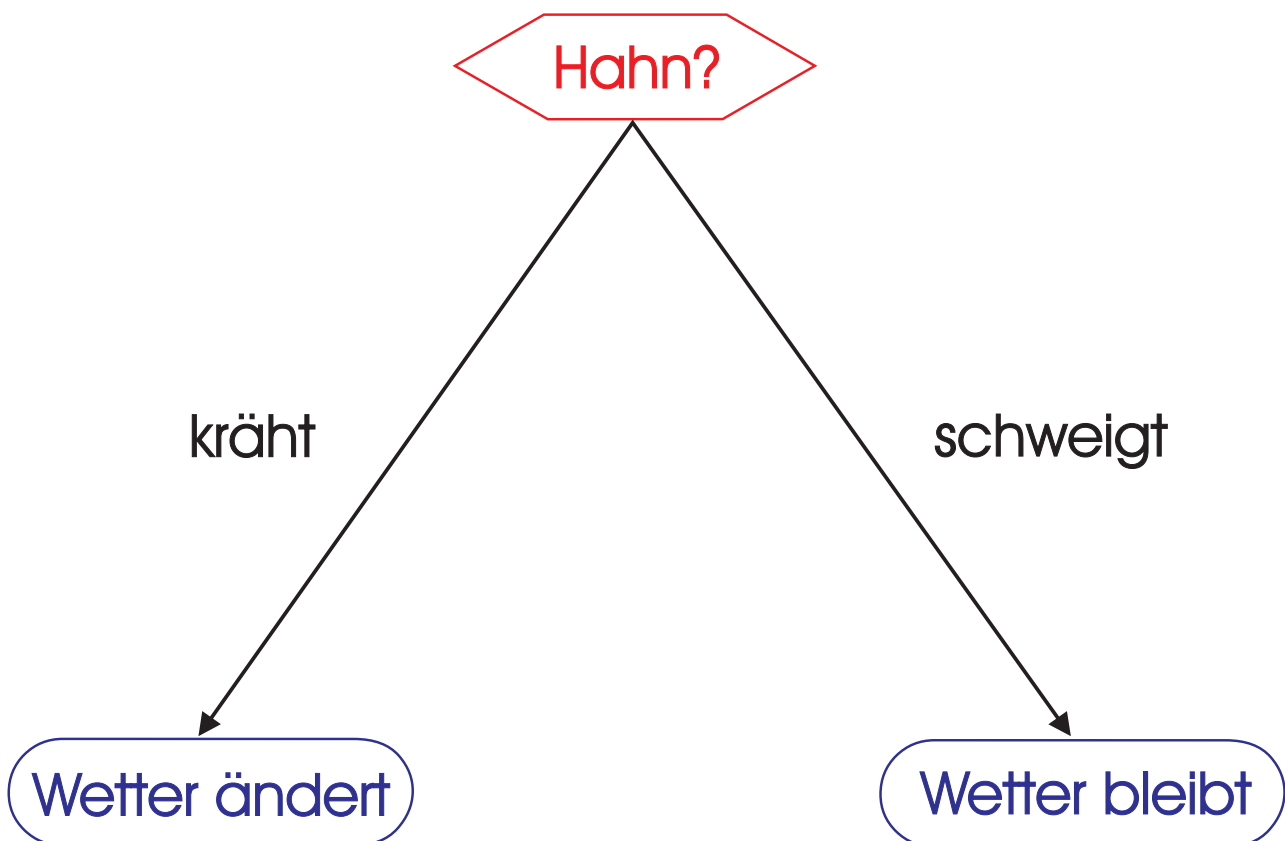
- Wertebereich { wahr, falsch }
- Beispieloperation: oder-Verknüpfung



## 4.8 Beispiel eines Entscheidungsbaums

WENN **der Hahn** kräht  
DANN ändert sich das Wetter

WENN **der Hahn** schweigt  
DANN bleibt das Wetter



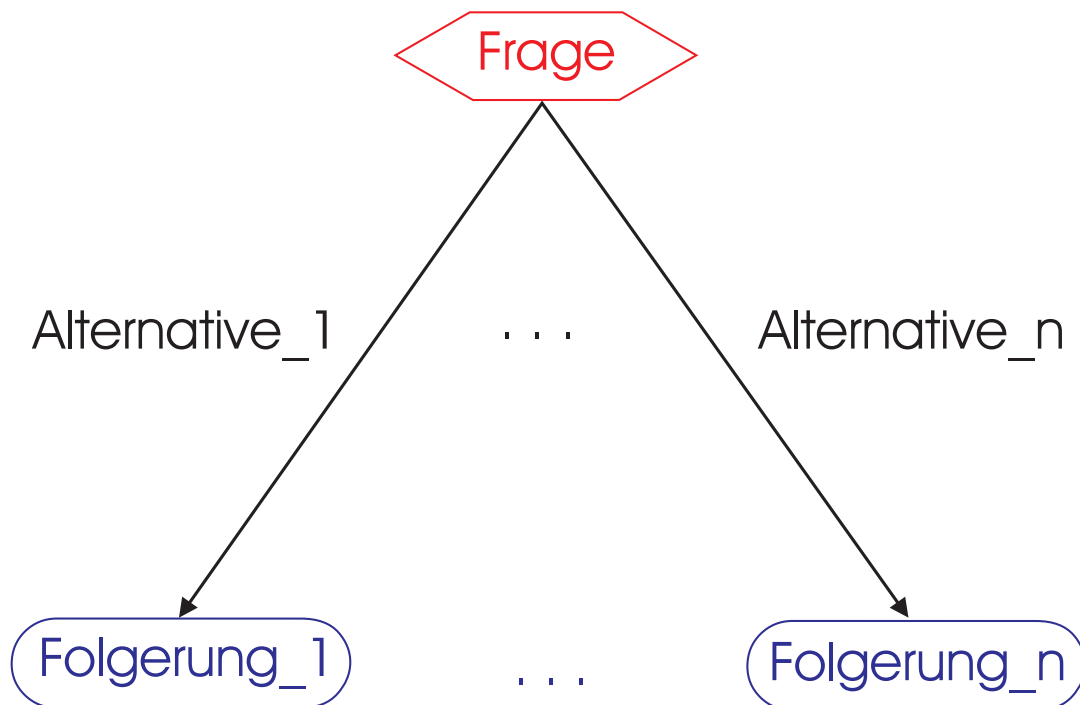
Ein **Entscheidungsbaum** fasst *mehrere* Regeln in einem einzigen Baum zusammen

## 4.9 Ein einstufiger Entscheidungsbaum

WENN **Frage** Alternative\_1  
DANN **Folgerung\_1**

...

WENN **Frage** Alternative\_n  
DANN **Folgerung\_n**



Eine **Folgerung** ist entweder ...

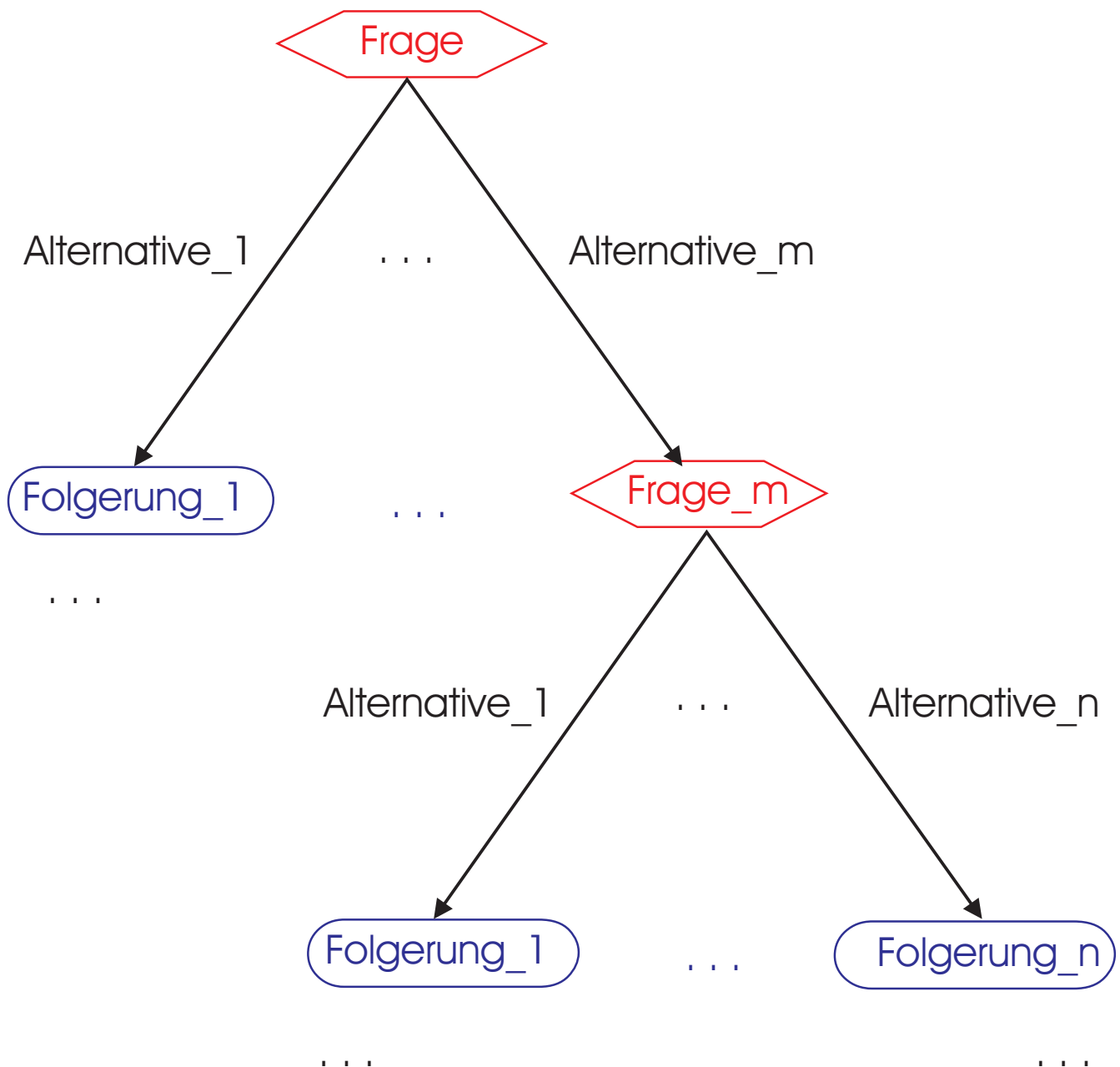
*(Attribut, Wert) oder ...*

*Aktion oder ...*

*Entscheidungsbaum*

## 4.10 Ein zusammengesetzter Entscheidungsbaum

Die Verknüpfung **einstufiger** Entscheidungsbäume ergibt einen **zusammengesetzten** Entscheidungsbaum.

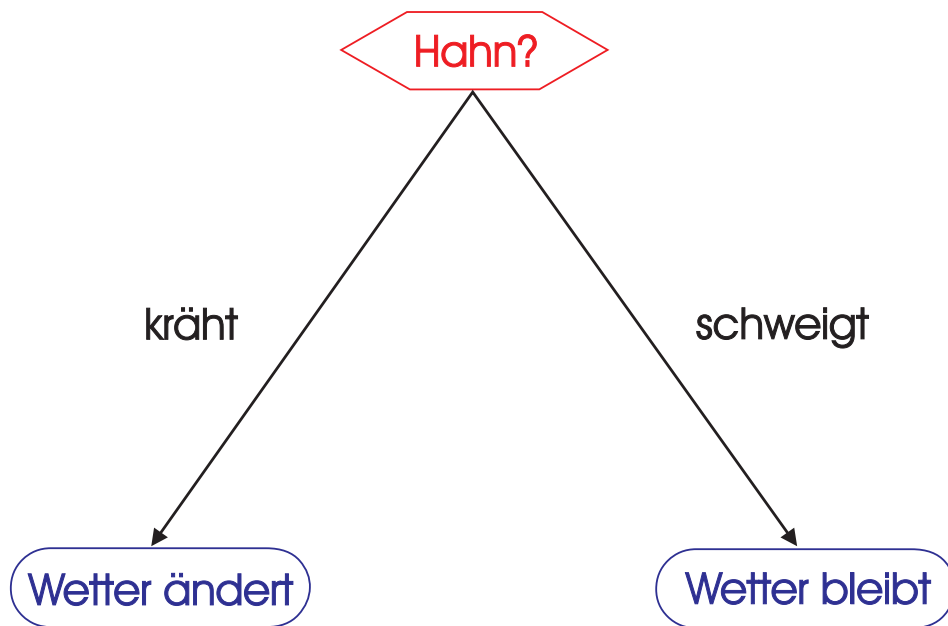


# Terminologie

**Entscheidungsbaum** (Klassifikationsbaum) :=  
Hierarchie von  
Regeln mit Bedingungen,  
deren geordnete Beantwortung  
einem Klassifikationsobjekt  
eine von mehreren Folgerungen zuordnet

<i>Regelterminologie</i>	<i>Baumterminologie</i>	<i>Beispiel</i>
Regelmenge	Entscheidungsbaum	-
Regel	Ast	-
Bedingung (Attribut-Wert)	Knoten und Kante	<i>Hahn kräht</i>
Bedingungsattribut (Frage)	Wurzel-/Zwischenknoten	<i>Hahn?</i>
Bedingungswert (Alternative)	Kante	<i>kräht</i>
Folgerung (Attribut-Wert oder Aktion)	Blattknoten	<i>Wetter ändert</i>

# Entscheidungsbaum als Einrückungsliste



Hahn?  
kräht  
    Wetter ändert  
schweigt  
    Wetter bleibt

Frage  
    Alternative\_1  
        Folgerung\_1  
    ...  
    Alternative\_n  
        Folgerung\_n

---

# Entwicklung regelbasierter Expertensysteme

---

## Grundlagen

Beispiele <i>Künstlicher Intelligenz</i>	<u>6</u>
Schwierigkeiten	<u>8</u>
Expertensysteme	<u>10</u>
Wissen als Regeln darstellen	<u>21</u>

## Entwicklung mit *XpertRule*

⇒ Spesen 	<u>31</u>
⇒ Betriebskredit	<u>45</u>
⇒ Wissen erklären	<u>49</u>

## Theorie

• Einfache Regeln darstellen	<u>56</u>
• Schlüsse aus einfachen Regeln ableiten	<u>59</u>
• Regelverkettung mit <i>VBA</i>	<u>62</u>

## 4.12 📌 SPESEN - Fallbeispiel

*Ein Expertensystem SPESEN soll abklären, ob die Spesenabrechnung eines Mitarbeiters in Ordnung ist. Der Bescheid hängt von den folgenden Faktoren ab :*

*Funktion des Antragstellers (Bsp. Direktor)*

*Klasse des gebuchten Hotels (Bsp. Economy)*

*Abteilungszugehörigkeit des Antragstellers*

- ⇒ Feinspezifikation
- Aufrufhierarchie der Regelmodule
- Regelmodule
- Beispielsitzung
- Entscheidungsbäume

---

## 4.12 SPESEN - ① Feinspezifikation

---

### Regelmodul *Spesenbescheid*

Ein Direktor erhält die Spesen immer bezahlt. Der Spesenbescheid für einen Vizedirektor ist hingegen nur dann positiv, falls die Hotelklasse "Standard" ist und der Vizedirektor der Abteilung "Rechnungswesen" angehört. Prokuristen erhalten die Spesen nur dann erstattet, falls sie ein "Economy"-Hotel gebucht haben.

### Regelmodul *Hotelklasse*

Wenn die Hotelspesen nach Steuern kleiner als 50 sind, dann gehört das Hotel zur Klasse "Economy". Zürcher Hotels mit Kosten zwischen 50 und kleiner 100 gehören zur Klasse "Standard". Hotels der gleichen Preisklasse, die ausserhalb von Zürich liegen, zählen wir aber zur Klasse "Komfort". Ebenfalls zu dieser Klasse gehören Hotels, die mehr als 100 kosten.

### Regelmodul *SpesenantragVollständig*

Falls die Hotelleistungen mindestens die Ausgaben für "Zimmer" und "Mahlzeiten" umfassen, berechnen wir für Buchhaltungszwecke noch die Spesen vor Steuern und veranlassen dann die Auszahlung der Spesen mit Steuern. Ist die Spesenabrechnung unvollständig, veranlassen wir eine Rückfrage beim Antragsteller.

### *Aufgabe*

Stellen Sie die drei Texte als Regeln der folgenden Form dar :

WENN

*Attribut* <Vergleichsoperator> *Wert* UND (bzw. ODER)

...

DANN

*Attribut* <Vergleichsoperator> *Wert* (bzw. *Aktion*)



---

## SPESEN - ② Aufrufhierchie der Regelmodule

---

Der Übersichtlichkeit halber teilen wir das Expertensystem SPESEN in drei *Regelmodule* ein. Jedes besteht aus lokalen *Attributen* und einem *Entscheidungsbaum* und ruft direkt oder indirekt die eingerückten Regelmodule:

### Spesenbescheid

lokale Attribute

Entscheidungsbaum →

#### Hotelklasse

lokale Attribute

Entscheidungsbaum

#### SpesenantragVollständig?

lokale Attribute

Entscheidungsbaum

Feinspezifikation

Aufrufhierarchie der Regelmodule

⇒ Regelmodule

- Beispielsitzung
- Entscheidungsbäume

# Regelmodul<sup>1</sup>

**Regelmodul** (engl. task) :=

Teil eines Expertensystems, dessen Attribute und Regeln inhaltlich und formal zusammengehören

Regelmodule lassen sich darstellen ...

umgangssprachlich oder als ...

Regeltext oder als ...

graphischer Entscheidungsbaum

7

---

<sup>1</sup> Unterschied zwischen Expertensystemen und Programmen mit IF-THEN-ELSE ?

---

## 4.13 SPESEN - ③ Regelmodul *Spesenbescheid*

---

Wie lautet der *Spesenentscheid* ?

**WENN** Funktion = "Direktor"

**DANN** *SpesenantragVollständig?* % Dieses Modul ausführen

**WENN**

Funktion = "Vizedirektor" **und**

*Hotelklasse* = "Komfort" % Ergebnis von Modul *Hotelklasse*

**DANN** Spesenantrag ablehnen

**WENN**

Funktion = "Vizedirektor" **und**

*Hotelklasse* = "Standard" **und**

Abteilung = "Rechnungswesen"

**DANN** *SpesenantragVollständig?*

**WENN**

Funktion = "Vizedirektor" **und**

*Hotelklasse* = "Standard" **und**

Abteilung = "Verkauf"

**DANN** Spesenantrag ablehnen

**WENN**

Funktion = "Prokurist" **und**

*Hotelklasse* = "Komfort" **oder**

*Hotelklasse* = "Standard"

**DANN** Spesenantrag ablehnen

**WENN**

Funktion = "Prokurist" **und**

*Hotelklasse* = "Economy"

**DANN** *SpesenantragVollständig?*

...

---

## 4.14 / 4.15 SPESEN - Regelmodule *Hotelklasse*, ...

---

Wie heisst die *Hotelklasse*?

### WENN

SpesenMitSteuern < 50

**DANN** Hotelklasse = "Economy"

### WENN

SpesenMitSteuern >= 50 **und**

SpesenMitSteuern < 100 **und**

Hotel in Zürich

**DANN** Hotelklasse = "Standard"

### WENN

SpesenMitSteuern >= 50 **und**

SpesenMitSteuern < 100 **und**

Hotel nicht in Zürich

**DANN** Hotelklasse = "Komfort"

### WENN

SpesenMitSteuern >= 100

**DANN** Hotelklasse = "Komfort"

Ist der *SpesenantragVollständig*?

### WENN

Hotelleistungen = Obermenge von "Zimmer" und "Mahlzeiten"

### DANN

*BerechneSpesenVorSteuern*

% Diese Prozedur ausführen

Spesenantrag bezahlen


**SONST** Beim Antragsteller nachfragen

## 4.16 SPESEN - Beispielsitzung I

 **Spesenbescheid** X

Welche Funktion bekleidet der Antragsteller ?

Direktor  
**Vizedirektor**  
Prokurist

 **Welcher Klasse gehört das Hotel an ?** X

Wie hoch sind die Spesen inklusive Steuern ?



Value:

 **Welcher Klasse gehört das Hotel an ?** X

Liegt das Hotel in Zürich ?





**Ja**  
Nein



# SPESEN - Beispielsitzung II

 **Spesenbescheid** 

Welcher Abteilung gehört der Antragsteller an ?


**Rechnungswesen**  
Verkauf




   

 **Hotelleistungen** 









Welche Hotelleistungen haben Sie beansprucht ?

**Zimmer**  
**Mahlzeiten**  
Wäsche  
Telefon und Fax  
Übrige



 **SpesenantragVollstaendig**  

Der Spesenantrag wird bezahlt für die folgenden Leistungen: Zimmer  
And Mahlzeiten. Der Betrag ist 80 nach Steuern und 71 vor Steuern.



- ▶ *Hotelklasse* gibt Ergebniswert an *Spesenbescheid* (**Rückwärtsverarbeitung**)
- ▶ *Spesenbescheid* ruft *SpesenantragVollstaendig*, ohne einen Ergebniswert zu erwarten (**Vorwärtsverarbeitung**)

Feinspezifikation

Aufrufhierarchie der Regelmodule

Regelmodule

Beispielsitzung

Entscheidungsbäume

## SPESEN - Eigenschaften der Regelmodule

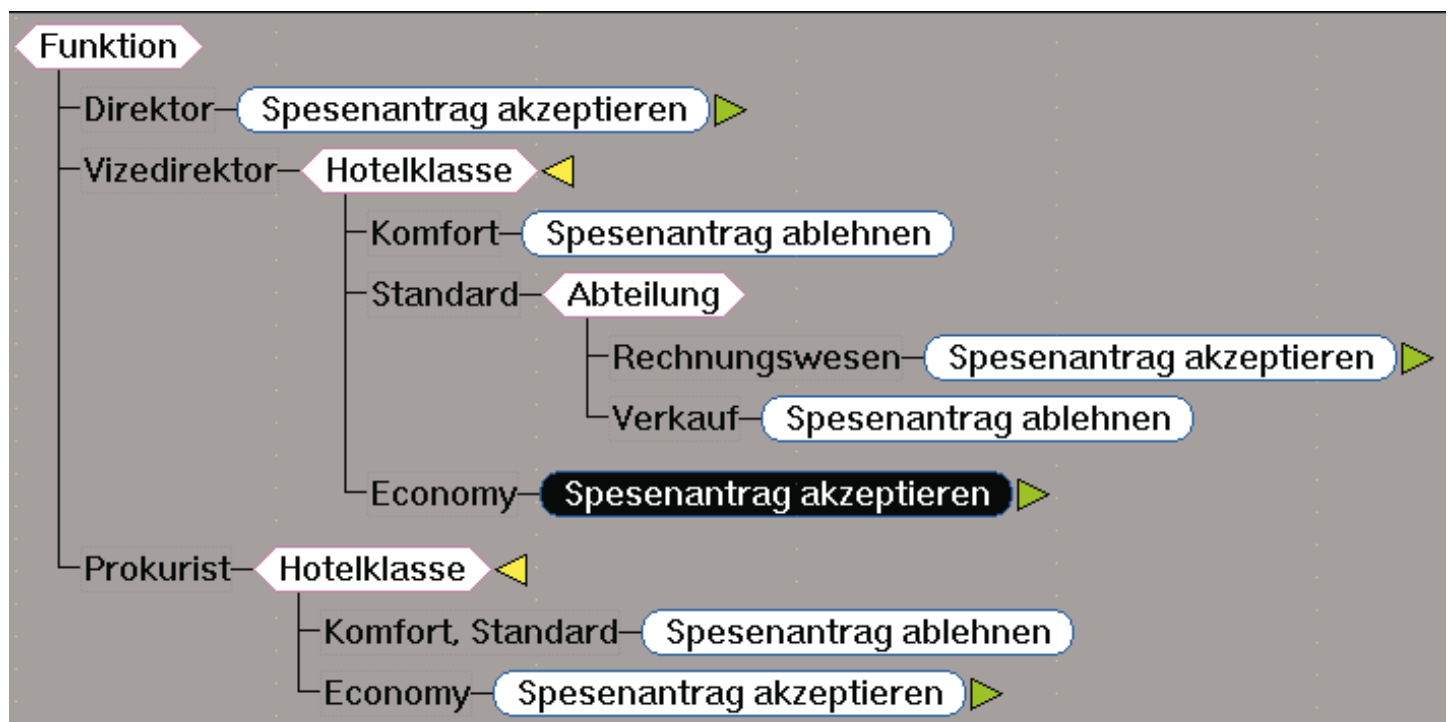
<i>Modul</i>	<i>Baum</i>	<i>Attribute</i>	<i>Aktion</i>	<i>Kommunikation mit Spesenbescheid</i>
<b><i>Spesenbescheid</i></b>				
<b><i>Hotelklasse</i></b>				rückwärts
<b><i>Spesenantrag- Vollständig</i></b>				vorwärts



## 4.18 SPESEN - Baum *Spesenbescheid*

- ◇ Bedingungsattribute
- Bedingungswerte
- Folgerungen

### Entscheidungsbaum<sup>1</sup>



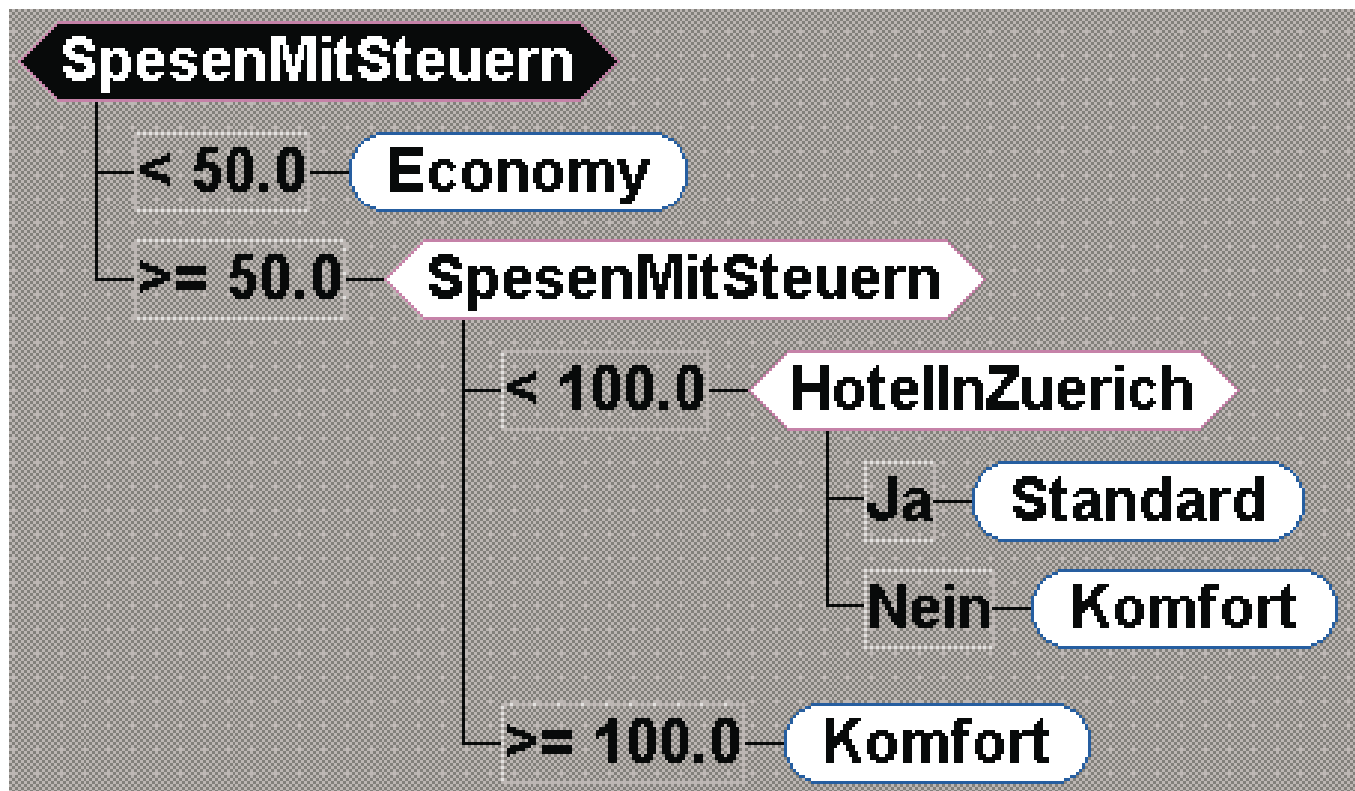
### Attribute

	Funktion	Abteilung	Hotelklasse	Spesenbescheid
	L	L	T L	T L Outcomes
1	Direktor	Rechnungswesen	Komfort	Spesenantrag akzeptieren
2	Vizedirektor	Verkauf	Standard	Spesenantrag ablehnen
3	Prokurist		Economy	

<sup>1</sup> Interpretieren Sie die Regeln und die Attribute verbal

# SPESEN - Entscheidungsbaum *Hotelklasse?*

## Entscheidungsbaum



## Attribute

	HotellInZuerich	SpesenMitSteuern	Hotelklasse
	<b>B</b>	<b>N0</b>	<b>T L Outcomes</b>
1	Ja		Komfort
2	Nein		Standard
3			Economy

**Bedingungsattribute** HotellInZuerich  
SpesenMitSteuern

**Folgerungsattribut** Hotelklasse

## 4.19 SPESEN - Baum SpesenantragVollständig?

### Entscheidungsbaum

#### SpesenMitSteuern

**P** Berechne Spesen Ohne Steuern

Hotelleistungen

Superset of Zimmer,  
Mahlzeiten

Bezahlen

Otherwise Nachfragen

### Attribute

	SpesenMitSteuern	Hotelleistungen	SpesenantragVollstaendig
	NO	LL	TL Outcomes
1		Zimmer	Bezahlen
2		Mahlzeiten	Nachfragen
3		Wäsche	
4		Telefon und Fax	
5		Übrige	

### Aktion

**P** Aktion BerechnungsProzedur

(SpesenOhneSteuern = SpesenMitSteuern / 1.12)

---

# Funktionalität von *XpertRule* KBS

---

## Eingabe (Wissenserwerb)


- explizit -
  - grafisch (Entscheidungsbaum) ←
  - textuell ←
- implizit -
  - ›Induktion mit Konsistenzprüfung →

## Verarbeitung (Wissensherleitung)

- Verarbeitungsrichtung
  - vorwärts →
  - rückwärts ←
- Prozeduren (Skriptsprache)

## Ausgabe

- Query the User → Beispielsitzung
- Erklärungen → Erklärungsdialog

Laden Sie  [Betriebskredit.xra](http://Betriebskredit.xra) und klicken Sie auf “Demo”. Sie können dann die geladene Anwendung analysieren und ändern. Änderungen lassen sich allerdings nicht speichern. Die Anwendung prüft die Bonität eines Kreditnehmers. Sie ist mit achtzehn Regelmodulen umfangreicher als das Expertensystem *Spesenabrechnung*. Einzelne Entscheidungsbäume enthalten ausserdem mehr als hundert Knoten.

### *Lernziele*

- ⇒ Modulhierarchie interpretieren
- ⇒ Query the User anwenden
- ⇒ Dictionary verwenden
- ⇒ Entscheidungsbäume interpretieren

Sie erhalten in XpertRule KBS auf vier Arten Hilfe:

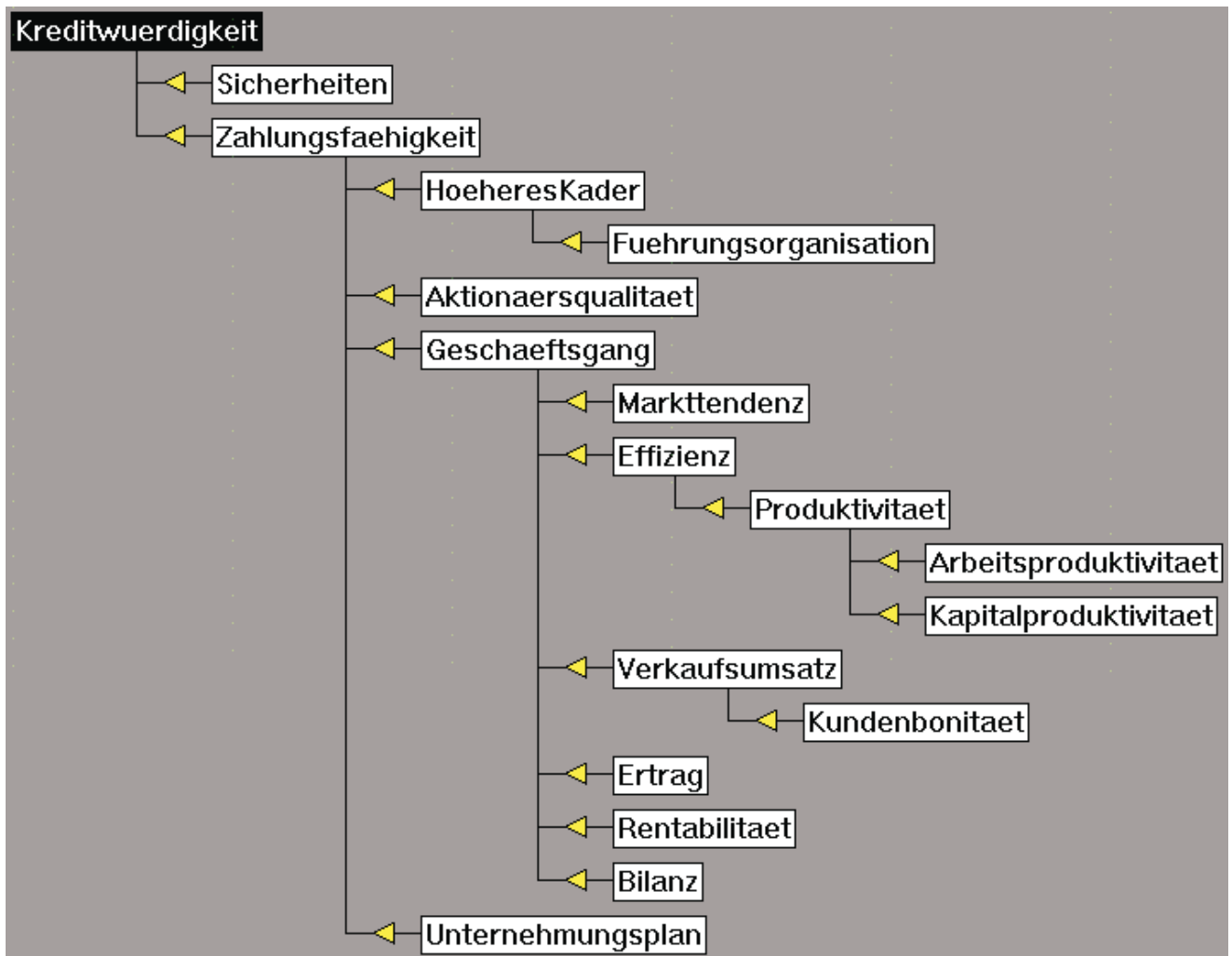
- Wenn Sie den *Cursor* auf ein Toolbar-Symbol positionieren, erscheint eine Kurzbeschreibung des Symbols.
- Ausführliche Hilfe erhalten Sie auf einem Menüpunkt mit *Shift/F1*.
- Ein Klick auf das Toolbar-Icon ? erläutert den laufenden Bildschirm.
- Der Menüpunkt “Help/User Guide/Tutorial” führt am Beispiel *Spesenabrechnung* in die Benutzung von XpertRule KBS ein.

### *Query the User*

- a) Nach dem Laden von: Betriebskredit.xra erscheint die Modulhierarchie 4.20. Testen Sie die Anwendung, indem Sie auf den Menüpunkt “Run” klicken und dann “from start task” wählen. Verwenden Sie während des Dialogs die Schaltflächen “Back” und “Again”, sobald Sie einen Schritt zurück oder wieder an die Wurzel des Entscheidungsbaums gelangen möchten.

## Regelmodule

b) Beschreiben Sie an der folgenden Abbildung den Aufbau des Expertensystems.



## Dictionary

c) Klicken Sie auf das Buchsymbol ("Dictionary") und gewinnen Sie einen groben Überblick über die Objekttypen der Anwendung:

### Dictionary

#### Attribute

Aufzählungsattribute (Zeichenketten, **L**ist)

Zahlen (**N**umeric)

#### Aktionen

Dialog (**G**UI)

Bericht (**R**eport)

Prozedur (**P**rocedure)

In welchem Regelteil können Attribute bzw. Aktionen auftreten?

- d) Betrachten Sie ausgewählte Objekte, indem Sie zweimal darauf klicken. Schauen Sie sich zum Beispiel die Werte und den Dialog zum Attribut *Bankbeziehung* an.
- e) Eruieren Sie, welche Berichtobjekte (R) zu welchen Attributen gehören.

### *Entscheidungsbäume*

- f) Öffnen Sie die Modulhierarchie, indem Sie auf das Kettensymbol ("Chaining Map") klicken. Markieren Sie das Modul *Sicherheiten* und klicken Sie auf das Symbol "Decision Tree". Übersetzen Sie den gezeigten Entscheidungsbaum in umgangssprachliche WENN ... DANN-Regeln.
- g) (1) Unter welcher Voraussetzung ist das Ergebnis (engl. outcome) des Entscheidungsbaums *Sicherheiten* "wahrscheinlich"?  
(2) Das Modul *Sicherheiten* liefert dem Hauptmodul *Kreditwürdigkeit* den Wert der Variable *Sicherheiten* zurück. Welchen Wert nimmt die Variable an, wenn *SicherheitenLiquide* = "ja" und *Deckung* = "70 bis 100%" sind?  
(3) Prüfen Sie Ihre Antwort, indem Sie das Expertensystem vom Modul *Sicherheiten* aus starten ("Run").
- h) Betrachten Sie den Entscheidungsbaum *Zahlungsfähigkeit* und übersetzen Sie ihn in umgangssprachliche Regeln.
- i) Welche Regeln ziehen Sie aufgrund Ihres betriebswirtschaftlichen Wissens und Ihrer Praxiserfahrung in Zweifel?
- j) Viele Benutzereingaben lassen sich besser operationalisieren, wenn Aufzählungsattribute durch numerische Kennzahlen ersetzt werden. Ein wichtiger Gegenstand von Kennzahlen ist die Unternehmensrentabilität, zum Beispiel in der folgenden Form:

*Unternehmensrentabilität des Eigenkapitals (EK) =*  
$$\frac{(\text{Unternehmungserfolg} + \text{Zinsen auf dem EK})}{((\text{EK der Eröffnungsbilanz} + \text{EK der Schlussbilanz}) / 2)}.$$

Untersuchen Sie, in welchem Modul und wie Sie diese Kennzahl verwenden können.

- k) Das Expertensystem *Betriebskredit* bietet weitere Operationalisierungsgelegenheiten. Suchen Sie nach Attributen, die sich durch Kennzahlen der folgenden Liste ersetzen oder ergänzen lassen:

*Innere Investitionsdeckung* = Cash Flow / Neuinvestition

*Debitorenumschlag* = Warenertrag /  
$$((\text{Debitoren Eröffnungsbilanz} + \text{Debitoren Schlussbilanz}) / 2)$$

*Durchschnittliche Debitorenfrist* = 360 / Debitorenumschlag

*Fremdfinanzierungsgrad* = Fremdkapital / Gesamtkapital

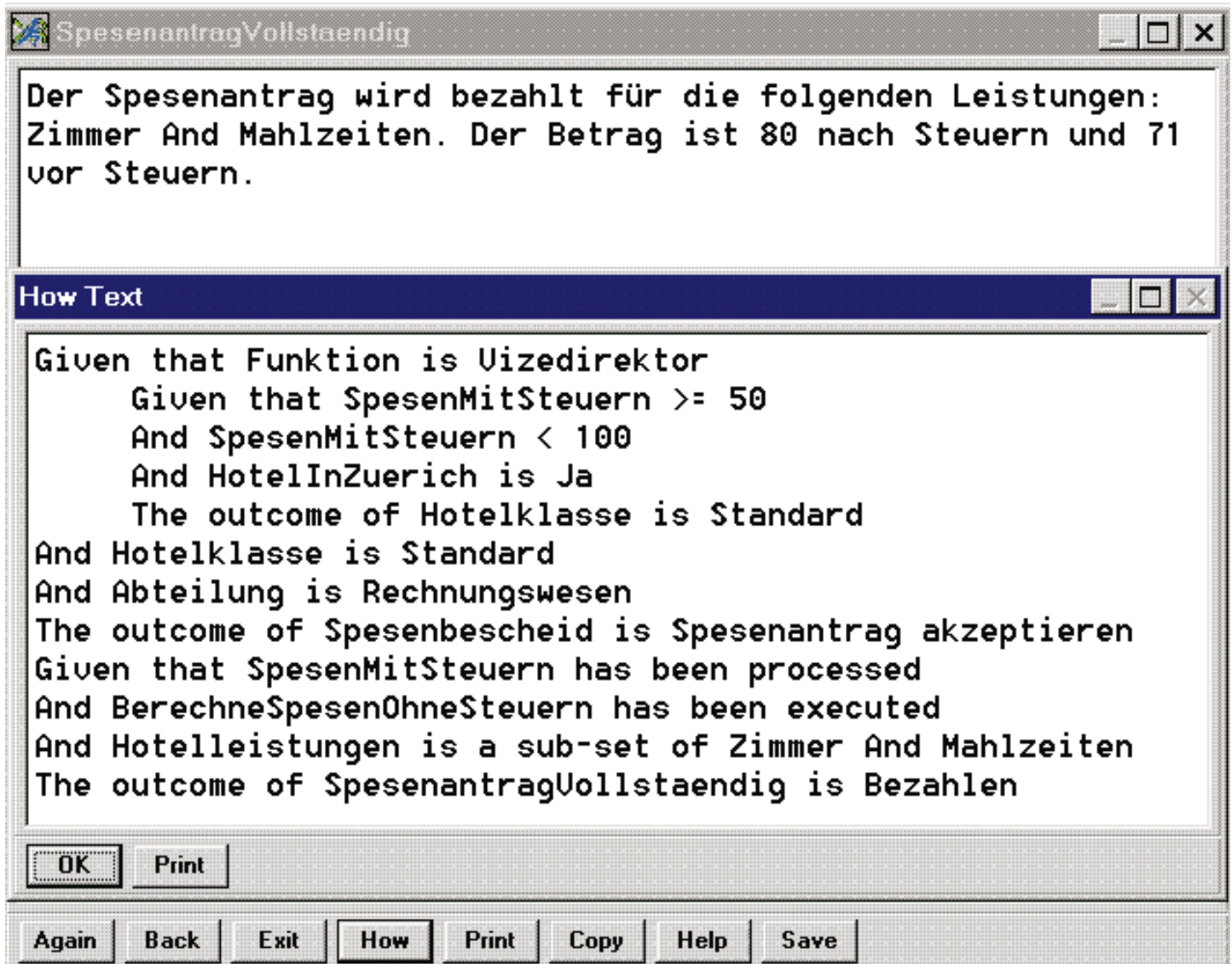
*Barliquidität* = flüssige Mittel / kurzfristiges Fremdkapital

*Einzugsliquidität* = (flüssige Mittel + kurzfristige Forderungen) /  
kurzfristiges Fremdkapital.

- l) Diskutieren Sie die Praxisrelevanz des Expertensystem-Ansatzes. Schauen Sie sich dazu die Fallbeispiele auf der Web Site von XpertRule an (<http://www.attar.com/pages/cases.htm>).



## 4.21 BETRIEBSKREDIT - Wie-Erklärung



## 4.22 Erklärungsarten

### Wissensbasis

$a1 \Leftarrow a2 \ \& \ a3$

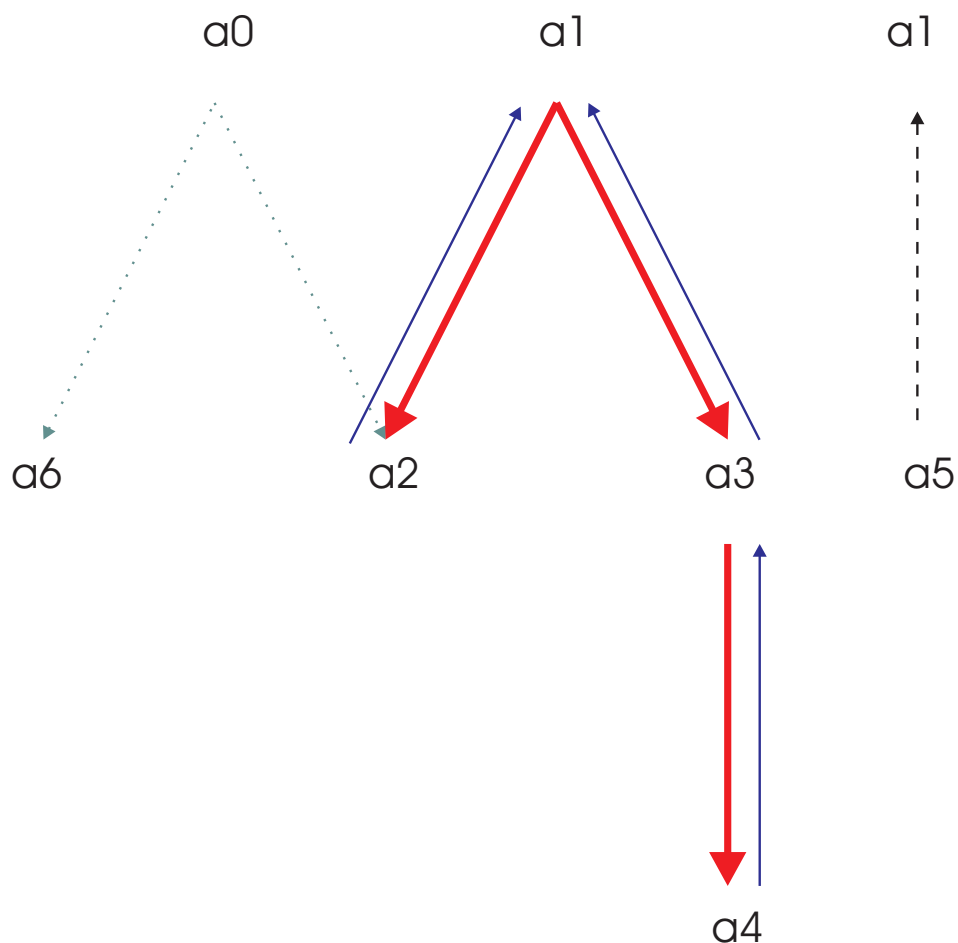
$a1 \Leftarrow a5$  (nur falls erste Regel scheitert)

$a2$  (Fakt)

$a3 \Leftarrow a4$  ( $a4$  werde durch *query the user* befragt)

$a0 \Leftarrow a2 \ \& \ a6$  ( $a6$  bewahrheitet sich nicht)

### Wissensherleitung



Wie | Warum | Warum nicht | Was-wenn

ExpertRule unterstützt nur Wie-Erklärungen

11

## 4.29 Betriebliche Anwendungen

### Anwendungen von XpertRule

(<http://www.attar.com/pages/cases.htm>)

Entwickler und Jahr	Zweck
<i>Department of Primary Industry and Fisheries of Tasmania</i> 1998	Anbau von Apfelbäumen
<i>M+W</i> (britische Ladenkette) 1997	Entdeckung von Ladendiebstahl
<i>Rubbers and Polymers Research Association</i> 1996	Verbreitung von Forschungsergebnissen
<i>Dialog</i> (schwedische IT-Beratung) 1995	Steuerberatung (Anwendung der Mehrwertsteuer)
<i>Norwich Union Life &amp; Pensions</i> (irische Versicherung) 1992	Abschluss von Lebensversicherungsverträgen
<i>Traversum AB</i> (schwedischer Börsenhändler) 1987	Wertschriftenhandel

---

## 4.28 Dynamik des Expertenwissens

---

*“We used Attar software to help develop real BPR solutions in our new business processing area. What began as a modest conceptual exercise has led us to develop new ways of processing **life proposals** and these methods now underpin a cost effective and service sensitive new business process. However, all of our systems and particularly service sensitive systems such as New Business Processing must be **dynamic**. We review and evaluate the experience from our Expert System Processing at least once a year. We do this to identify further niches to make the system more cost effective whilst maintaining or improving service and underwriting quality. Put simply, **without Expert System Processing we could not provide the quality service** needed to survive in today’s market whilst keeping the operation cost effective.”*

(Eoin Byrne, Operations Manager, Norwich Union Life in Ireland)

# Expertensystem-Werkzeuge?

<i><b>Wissen ...</b></i>	<i><b>Dediziertes Werkzeug</b></i>	<i><b>Prozedurale Sprache</b></i>
<i>darstellen</i>	syntaktisch einfach oft standardisiert	syntaktisch komplex programmier- und wartungsaufwendig
<i>erwerben</i>	benutzerfreundlich Validitätsprüfung eingebaut	programmieraufwendig oft Neukompilation
<i>herleiten</i>	eingebaut Methodenwahl	prozedural verknüpft mit Wissensdarstellung
<i>erklären</i>	eingebaute protokoll- abhängige Erklärungen	protokollabhängige Er- klärungen aufwendig
<i>kommunizieren</i>	eingebaut und erwei- terbar	programmieraufwendig

## 4.30 Methode im Vergleich

Kriterium	AHP	Optimierung	OLAP	Regelbasierte Systeme	Induktion	Neuronale Netze	Regression
Methode breit anwendbar	Ø	–	+	Ø <sup>1</sup>	Ø	Ø	–
Automatisierungsgrad	–	+	–	Ø <sup>2</sup>	+	+	+
Ergebnis genau	–	+	+	Ø	+	+	+
Unabhäng. Var. gewichtbar	–	–	–	–	Ø	–	+
Lösungsweg begründbar	Ø	–	Ø	+ <sup>3</sup>	+	–	–
Methode plausibel	+	Ø	+	+	Ø	–	Ø
Ergebnis einbettbar	–	+	Ø	Ø	+	Ø	+
Entwicklungsaufwand	+	+	–	– <sup>4</sup>	+	Ø	Ø
Rechnerbelastung	+	+	–	Ø	Ø	–	+

### Kriterien

- 1 vor allem für schlecht quantifizierbare Variablen
- 2 viele Benutzerinterventionen erforderlich
- 3 vor allem durch Wie-, Warum- und Warum nicht-Erklärungen
- 4 Hauptnachteil (Erleichterung durch automatische Regelinduktion)

### *Zusatznachteil*

Validierung schwierig

---

# Theoretischer Hintergrund

---

## Grundlagen

Beispiele <i>Künstlicher Intelligenz</i>	<u>6</u>
Schwierigkeiten	<u>8</u>
Expertensysteme	<u>10</u>
Wissen als Regeln darstellen	<u>21</u>

## Entwicklung mit *XpertRule*

Spesen 📌	<u>31</u>
Betriebskredit	<u>45</u>
Wissen erklären	<u>49</u>

## Theorie

⇒ Einfache Regeln darstellen	<u>56</u>
⇒ Schlüsse aus einfachen Regeln ableiten	<u>59</u>
⇒ Regelverkettung mit <i>VBA</i>	<u>62</u>

# Textuelle Regelnotationen

## ❶ Folgerung **rechts**

WENN  $b$  UND  $c$  DANN  $f$

## ❷ Folgerung **links** ( ❷ äquivalent ❶ )

$f$  WENN  $b$  UND  $c$

$f \Leftarrow b \ \& \ c$  ( $f$  gilt, wenn  $b$  und  $c$  gelten)

## ❸ Folgerung **zusammengesetzt**

WENN  $b$  DANN  $f$  UND  $g$

## Folgerung mit **Alternative** (    äquivalent    )

WENN  $b$  und  $c$

DANN  $f$

SONST  $g$

## Regel **geschachtelt** (statt zusammengesetzter Bedingung)

WENN  $b$

DANN

    WENN  $c$

        DANN  $f$

SONST  $g$

- ⇒ Eine einfache Wissensdarstellung
- Algorithmus zur Regelverkettung
- Implementation in *VBA*

Aufgabe: Übersetzen Sie von einer Notation in die andere



# Eine einfache Fakt- und Regelnotation

❶ **Fakt** ist eine wahre Aussage

$b$

❷ **Regel** verknüpft eine Folgerung mit ...

- Fakten oder
- weiteren Regeln

$a$                        $b$                       (  $a$  gilt, falls  $b$  gilt )

$c$                        $a$

❸ Voraussetzung ist einfach oder **zusammengesetzt**

$a$                        $b \ \& \ c$                       (  $a$  gilt, falls  $b$  und  $c$  gelten )

ODER wird durch Regeln mit gleicher Folgerung ausgedrückt

## 4.24 REGELVERKETTUNG - ① Wissensdarstellung

Fakten

b

c

Regeln

f  $\Leftarrow$  b & d & e

a  $\Leftarrow$  g & d

a  $\Leftarrow$  c & f

x  $\Leftarrow$  b

e  $\Leftarrow$  d

h  $\Leftarrow$  x & a

d  $\Leftarrow$  c

a  $\Leftarrow$  x & c

d  $\Leftarrow$  x & b

Frage

h ? (Lösen Sie die Aufgabe selbst!)

Eine einfache Wissensdarstellung

$\Rightarrow$  Algorithmus zur Regelverkettung

- Implementation in VBA

1. Vom unbekannten Ergebnis **h** aus **rückwärts** verketten
2. Bewiesene Zwischenergebnisse auf einen Stapel legen

1)  $f \Leftarrow b \ \& \ d \ \& \ e$

2)  $a \Leftarrow g \ \& \ d$

3)  $a \Leftarrow c \ \& \ f$  % zweites Vorkommen von a

4)  $x \Leftarrow b$

5)  $e \Leftarrow d$

6)  $h \Leftarrow x \ \& \ a$

7)  $d \Leftarrow c$

8)  $a \Leftarrow x \ \& \ c$  % drittes Vorkommen von a

9)  $d \Leftarrow x \ \& \ b$  % zweites Vorkommen von d

+ **h**

+ **a**

+ **f**

+ **e**

+ **d**

+ **x**

**c**

**b**

Stapel der **ursprünglichen** und abgeleiteten (+) Attribute

## 4.25 REGELVERKETTUNG - ③ Algorithmus

### Entwurfssprachlicher Algorithmus

```
ableitbar (Fragen)  
  FALLS leer (Fragen)  
    ableitbar := wahr EXIT  
  FÜR JEDE Regel  
    FALLS Kopf (Fragen) = Regel.Folgerung  
      Restfragen := ohneKopf (Fragen)  
      Restfragen := Regel.Bedingungen + Restfragen  
      FALLS ableitbar (Restfragen)  
        ableitbar := wahr EXIT  
  ableitbar := falsch
```

### Erklärung

Entwurfssprache	Beschreibung
Regel. <i>Folgerung</i>	Folgerung einer Regel
Regel. <i>Bedingungen</i>	Liste aller Bedingungen der Regel
<b>[a, b, c]</b>	Beispiel einer Fragenliste
<i>leer</i> ( <b>Fragen</b> )	Alle Fragen sind geprüft.
<i>Kopf</i> ( <b>Fragen</b> )	erste Frage einer Fragenliste
<i>ohneKopf</i> ( <b>Fragen</b> )	Rest einer Fragenliste

**ableitbar** (**Fragen**)

FALLS *leer* (**Fragen**)

ableitbar := wahr EXIT

FÜR JEDE Regel

FALLS *Kopf* (**Fragen**) = Regel.*Folgerung*

Restfragen := *ohneKopf* (**Fragen**)

Restfragen := Regel.*Bedingungen* + Restfragen

FALLS **ableitbar** (Restfragen)

ableitbar := wahr EXIT

ableitbar := falsch

## ① Test-Wissensbasis

a  $\Leftarrow$  b & c      % Regel mit Folgerung und zwei Bedingungen

b                    % “Regel” mit Folgerung ohne Bedingungen (Fakt)

c

## ② Werte, die **Fragen** hintereinander annimmt

a

b, c

c

leere Fragenliste

## ③ Rekursive Verkettung


**ableitbar** wird wahr, falls sich eine gegebene **Fragen**liste aus einer gegebenen Wissensbasis ableiten lässt

Mit dieser Aufgabe sollen Sie die Implementation eines Regelinterpreters nach dem Muster des Entwurfscode 4.25 *nachvollziehen*. Im ersten Teil interpretieren Sie die Benutzeroberfläche, im zweiten Teil interpretieren Sie den Programmcode von Visual Basic unter Excel.

### *Lernziele*

- ⇒ Benutzeroberfläche eines einfachen Regelinterpreters benutzen
- ⇒ Aufbau und Code der Benutzeroberfläche verstehen
- ⇒ Aufbau und Code des Regelinterpreters verstehen


### *Start*

 [Regelverkettung.xls](#) übersetzt den entwurfssprachlichen Verkettungsalgorithmus in eine VBA-Funktion unter MS Excel. Alt/F11 wechselt zwischen der Benutzeroberfläche und dem VBA-Code. (Abschnitt 4.5.2 geht näher auf die Implementation des VBA-Code ein). Falls Sie direkt von der CD ROM oder vom Server laden, speichern Sie die Datei mit “File/Save As ...” auf Ihre Festplatte (zum Beispiel auf D:\).



Beantworten Sie die folgenden Fragen zur Benutzer- und zur Entwicklersicht:

### *Benutzersicht*

- a) Versuchen Sie, die Aufgaben der Benutzeroberfläche zu erraten, ohne dass Sie auf die Schaltflächen klicken (Wenn nötig bewegen Sie den Cursor auf die Zelle “Hilfe” oder einen anderen Zellenkommentar). Was steht in der grünen, der blauen und der weissen Spalte? Was bezwecken die Schaltflächen “Beispiel”, “Starten” und “Löschen”?
- b) Analysieren Sie die Wissensbasis und beantworten Sie die Frage h?, ohne dass Sie das Programm bereits starten.

-  c) Klicken Sie auf “Beispiel” und starten Sie mit der Frage “h”.
- d) Wie kommen die Ausgaben der Spalte “Bewiesen” zustande?

### *Entwicklersicht*

- e) Welche Aufgaben erfüllen die drei Ereignisprozeduren, die der Code enthält? Beantworten Sie die Frage, ohne im Programm nachzuschlagen.
- f) Starten Sie den Programmeditor (Alt/F11) und suchen Sie die Ereignisprozeduren. Weshalb hat der Entwickler alle Prozedurnamen mit `_nachKlick` beendet?
-  g) Welche Aufgabe erfüllt `Starten_nachKlick()`?
- h) Verfolgen Sie den Programmablauf von `Starten_nachKlick()` Ordnen Sie dabei Tabellenblatt und Editor so an, dass Sie die Programmausgabe und den Code miteinander verfolgen können:
- Setzen Sie den Cursor auf `Starten_nachKlick()`.
  - Starten Sie den Einzelschrittmodus (F8).
  -  • Gehen Sie schrittweise durch die Prozedur, ohne die aufgerufenen Unterprogramme zu betreten (Umschalt/F8).
- i) Untersuchen Sie die Funktion `ableitbar(Frage)`.
- Testen Sie den Entwurfscode `ableitbar` von Abb. 4.25 an der (grösseren) Wissensbasis 4.24.
  - Vergleichen Sie die Funktion `ableitbar(Frage)` mit der entwurfssprachlichen Entsprechung. Achten Sie besonders auf die Implementation der Fragen- und Regellisten.
- j) Verfolgen Sie den Programmablauf von `ableitbar`:
- Setzen Sie einen Haltepunkt auf `ableitbar` (Taste F9).
  - Gehen Sie zurück zur Benutzeroberfläche (Alt/F11) und klicken Sie auf “Starten”. Sie können nun schrittweise durch `ableitbar` gehen (Umschalt/F8). Wenn Sie den Cursor auf einer Variable ruhen lassen, erhalten Sie den Inhalt angezeigt.

# Expertensysteme - Aufgabe zur Regelverkettung

Hilfe	Folienkapitel		Beispiel	Starten	Löschen
Folgerung	Bedingungen		Bewiesen		
f	b	d	<div>Microsoft Excel</div> <p>Welche Frage soll bewiesen werden?</p> <div>OK</div> <div>Abbrechen</div> <div>h</div>		
a	g	d			
a	c	f			
x	b				
e	d				
h	x	a			
d	c				
a	x	c			
d	x	b			
c					
b					



# Expertensysteme - Aufgabe zur Regelverkettung

Hilfe	Folienkapitel	Beispiel	Starten	Löschen
Folgerung	Bedingungen			Bewiesen
f	b	d	e	f
a	g	d		a
a	c	f		x
x	b			e
e	d			h
h	x	a		d
d	c			
a	x	c		
d	x	b		c
c				b
b				

**Ereignisprozeduren** := Prozeduren, die durch ein Ereignis (oft einen Klick auf eine **Schaltfläche**) aufgerufen werden

## Schaltflächen

**Beispiel**

aktiviert die voreingestellte Wissensbasis

**Starten**

liest eine Frage und ruft `ableitbar(Frage)`

**Löschen**

löscht Wissensbasis und bewiesene Aussagen

Codiere **Ereignisprozeduren** mit den Namen **Beispiel**, **Starten** und **Löschen**

Programmiersprachen erlauben oft nur eine mangelhafte Übersetzung des Entwurfscodes, vor allem wegen inadäquater ...

- **Datenstrukturen** (Objekte) oder
- **Aktionen** (Methoden)

*Visual Basic für Applikationen 5* akzeptiert zum Beispiel keine Arrays (Vektoren) als Funktionsargumente

Die folgende **VBA** - Funktion unterscheidet sich deshalb geringfügig vom **Entwurfscodes**

# REGELVERKETTUNG - Entwurf und VBA-Code

## Entwurfssprachliche Version

Entwurfssprache	Beschreibung
Regel.Folgerung	Folgerung der Regel
Regel.Bedingungen	Liste aller Bedingungen der Regel
[a, b, c]	Beispiel einer Fragenliste
leer (Fragen)	Fragen abgearbeitet
Kopf (Fragen)	erste Frage einer Fragenliste
ohneKopf (Fragen)	Rest einer Fragenliste

## VBA-Version

VBA	Beschreibung
Regeln.Rows	Regeln ist ein Tabellenblatt-Range ( For Each Regel In Regeln.Rows )
Regel.Cells (1)	Folgerung = 1. Zelle der If. Regel
Function Bedingungen(Regel As Range) As String	Regel.Bedingungen
"abc"	Fragenstring statt Fragenliste
""	leer (Fragen)
Left (Fragen, 1)	Kopf (Fragen)
Right (Fragen, Len (Fragen) - 1)	ohneKopf (Fragen)

## 4.27 REGELVERKETTUNG - Hauptfunktion in VBA

**ableitbar** (**Fragen**)

FALLS *leer* (**Fragen**)

ableitbar := wahr EXIT

FÜR JEDE Regel

FALLS *Kopf* (**Fragen**) = Regel.*Folgerung*

Restfragen := *ohneKopf* (**Fragen**)

Restfragen := Regel.*Bedingungen* + Restfragen

FALLS **ableitbar** (Restfragen)

ableitbar := wahr EXIT

ableitbar := falsch

Function **ableitbar** (ByVal **Fragen** As String) As Boolean

Dim Regel As Range

Dim lfFrage, Restfragen As String

If **Fragen** = "" Then

ableitbar = TRUE

Exit Function

End If

For Each Regel In Regeln.Rows

lfFrage = Left(**Fragen**, 1)

If lfFrage = *Regel.Cells*(1) Then

Restfragen = Right(**Fragen**, Len(**Fragen**) - 1)

Restfragen = *Bedingungen*(Regel) & Restfragen

If **ableitbar**(Restfragen) Then *'Rekursion*

ableitbar = TRUE

Exit Function

End If

End If

Next

ableitbar = FALSE

End Function

 [Web Quiz](#)

# *Folienverzeichnis (Ein Klick führt zur gewünschten Folie)*

<u>KI - Ein gescheiterter Traum ?</u>	<u>2</u>
<u>Einordnung</u>	<u>3</u>
<u>Unterrichtsmaterial</u>	<u>4</u>
<u>Grundlagen</u>	<u>5</u>
<u>Beispiele der Künstlichen Intelligenz</u>	<u>6</u>
<u>Einige Anwendungen der Künstlichen Intelligenz</u>	<u>7</u>
<u>Löst die KI praktische Probleme?</u>	<u>8</u>
<u>Ungelöste Probleme an der Sprachübersetzung</u>	<u>9</u>
<u>† Reparaturassistent - Ein Expertensystem</u>	<u>10</u>
<u>Begriff des Expertensystems</u>	<u>11</u>
<u>Expertensysteme sind meist regelbasiert:</u>	<u>12</u>
<u>Erklärungen in regelbasierten Systemen</u>	<u>13</u>
<u>4.1 Daten - und Wissensverarbeitung</u>	<u>14</u>
<u>Numerische und symbolische Verarbeitung</u>	<u>15</u>
<u>Expertensysteme - Eine Definition</u>	<u>16</u>
<u>† Leasing oder Kreditkauf - Ein Expertensystem</u>	<u>17</u>
<u>4.5 Architektur</u>	<u>18</u>
<u>4.7 Wissensklassen</u>	<u>19</u>
<u>Expertenwissen darstellen und herleiten</u>	<u>20</u>
<u>Wissen als Regeln darstellen</u>	<u>21</u>
<u>Regel aus Bedingung und Folgerung</u>	<u>22</u>
<u>Attribut, Wert und Datentyp</u>	<u>23</u>

<u>Datentyp</u>	24
<u>4.8 Beispiel eines Entscheidungsbaums</u>	25
<u>4.9 Ein einstufiger Entscheidungsbaum</u>	26
<u>4.10 Ein zusammengesetzter Entscheidungsbaum</u>	27
<u>Terminologie</u>	28
<u>Entscheidungsbaum als Einrückungsliste</u>	29
<u>Entwicklung regelbasierter Expertensysteme</u>	30
<u>4.12  SPESEN - Fallbeispiel</u>	31
<u>4.12 SPESEN - Feinspezifikation</u>	32
<u>SPESEN - Aufrufhierchie der Regelmodule</u>	33
<u>Regelmodul</u>	34
<u>4.13 SPESEN - Regelmodul <i>Spesenbescheid</i></u>	35
<u>4.14 / 4.15 SPESEN - Regelmodule <i>Hotelklasse</i>, ...</u>	36
<u>4.16 SPESEN - Beispielsitzung I</u>	37
<u>SPESEN - Beispielsitzung II</u>	38
<u>4.17 SPESEN - Kommunikation zwischen Modulen</u>	39
<u>SPESEN - Eigenschaften der Regelmodule</u>	40
<u>4.18 SPESEN - Baum <i>Spesenbescheid</i></u>	41
<u>SPESEN - Entscheidungsbaum <i>Hotelklasse</i>?</u>	42
<u>4.19 SPESEN - Baum <i>SpesenantragVollständig</i>?</u>	43
<u>Funktionalität von <i>XpertRule</i> KBS</u>	44
<u>__BETRIEBSKREDIT mit <i>XpertRule</i> KBS (A 4.1)</u>	45
<u>4.21 BETRIEBSKREDIT - Wie-Erklärung</u>	49
<u>4.22 Erklärungsarten</u>	50

4.29 Betriebliche Anwendungen	51
4.28 Dynamik des Expertenwissens	52
Expertensystem-Werkzeuge?	53
4.30 Methode im Vergleich	54
Theoretischer Hintergrund	55
Textuelle Regelnotationen	56
Eine einfache Fakt- und Regelnotation	57
4.24 REGELVERKETTUNG - Wissensdarstellung	58
REGELVERKETTUNG - Einfacher Regelverketter	59
REGELVERKETTUNG - Verkettungsbeispiel	60
4.25 REGELVERKETTUNG - Algorithmus	61
REGELVERKETTUNG mit VBA (A 4.3)	62
4.26 REGELVERKETTUNG - Benutzeroberfläche	64
REGELVERKETTUNG - Oberfläche nachher	65
REGELVERKETTUNG - Ereignisprozeduren	66
REGELVERKETTUNG - Codierung in VBA	67
4.27 REGELVERKETTUNG - Hauptfunktion in VBA	68