

Raimond Reichert

# Entscheidungstabellen als Kommunikationsmittel

# Inhalt

Motivation

Anwendungsbereich

Modellierung

Beispiele

# Motivation

## Ursprung der Entscheidungstabelle

Tabular forms for computer programming dates back to the late 1950's when General Electric, the Sutherland Corporation, and the United States Air Force worked on a complex file maintenance project. **Attempts at using flowcharts and traditional narratives failed to define the problem after more than six labor-years of effort.** This was typical of most large projects at that time. Then, **in 1958 four analysts using decision tables successfully defined the problem in less than four weeks.** When the right tool was used the problem was solved almost immediately.

**1958**

**Vier Analysten, vier Wochen,  
eine Entscheidungstabelle  
erfolgreich**

**1950er  
Sechs Mannjahre  
klassische Spezifikation  
ohne Erfolg**

**General Electric  
Sutherland Corporation  
United States Air Force**



# Ein Weg zu Entscheidungstabellen

## Tabellen plus diverse Erklärungen...

Existing Access							
	Whitelist aktiviert	Whitelist	Counter	Speed	Product Exclusion Rules	Action	Counter inc.
<b>Shop</b>							
Shop	1	0	x	x	x	TDM CasaTrio Bestellcase	0
Shop	1	x	0	x	x	TDM CasaTrio Bestellcase	0
Shop	1	x	x	0	x	TDM CasaTrio Bestellcase	0
Shop	1	x	x	x	0	TDM CasaTrio Bestellcase	0
Shop	1	1	1	1	1	Forward auf All-IP Adoption mit brokerId	+1

### Shop daVinciDuo (Counter hat keine Relevanz, wird nie inkrementiert)

Shop daVinciDuo	1	16 All-IP
Shop daVinciDuo	1	CR248 -
Shop daVinciDuo	1	Erweiterung P-
Shop daVinciDuo	1	Stack / N-Stack
		Weiche
		Selfcare

### Shop daVinciTrio (Counter hat keine Relevanz, wird nie inkrementiert)

Shop daVinciTrio	1
Shop daVinciTrio	1
Shop daVinciTrio	1
Shop daVinciTrio	1

### Endkunde (Whitelist nur d)

Endkunde	1
Endkunde	1
Endkunde	0
Endkunde	0
Endkunde	0
Endkunde	0
Endkunde	0

Die eOrders N-/P-Stack Weiche in eOrders muss für den Bereich Selfcare wie folgt funktionieren:

- Bei einer Casa Trio Bestellung auf einen neuen Anschluss (Activation) muss eOrders direkt auf den N-Stack weiterleiten
- Bei einer Casa Trio Bestellung auf einen bestehenden Anschluss (Adoption) muss eOrders den Kunden durch die Weiche schicken.

Innerhalb der Weiche muss eOrders die folgenden Punkte überprüfen:

- All-IP spezifischer LQS Check (Tibco), ob der Kunde mindestens 100 MBit/s Downstream hat
- Die Kundensprache dem oder der entsprechend in der Weiche hinterlegten zugelassenen Sprachen entspricht
- Der N-Stack Routing Zähler die im Konfigurationsprofil eOrders festgelegene Grenze nicht überschritten hat
- Der Service ValidateAdoption von Casa Trio auf dem N-Stack aktiviert

Sind alle die oben erwähnten Kriterien erfüllt (true), dann muss eOrders den Kunden im Falle einer Adoption auf den N-Stack weiterleiten.

Ist die Gesamtaussage aller Kriterien false, dann muss eOrders den Kunden auf dem P-Stack mit Casa Trio bedienen.

Zusatzbedingung:

- Ist der Konfigurationsparameter (Profil eOrders-Selfcare-test) aufgeschaltet, so erfolgt das Routing im Falle einer Adoption auf den N-Stack.

LQS Check:

- Für Release 1.1 (Mai 2010) gilt, dass der LQS Check über den bestehenden LQS-Service aufgerufen wird.
- Für Release > 1.1 gilt, dass der LQS Check über Tibco aufgerufen werden muss.

**Nur ein Auszug der Spezifikation**

# Ein Weg zu Entscheidungstabellen ... führen zu unwartbarem Code

```
public RoutingDecision getTrioBundleActivationRoutingDecision(final UserType userType, final String brokerId,
    final boolean checkAndIncrementCounter, final String customerLanguage) {
    if (userType.isShop()) {
        // Important: Ensure that incrementAllIpCounter is called as the last argument as this call increments the
        // counter at the same time as returning whether the increment was successful or not.
        if (configHelper.isLanguageSupported(customerLanguage) && isShopOnWhitelist(brokerId)
            && (!checkAndIncrementCounter || checkAndIncrementAllIpCounter())) {
            return RoutingDecision.ACTIVATION_ALL_IP;
        } else {
            return RoutingDecision.ACTIVATION_TDM_SHOP_AGENT;
        }
    } else if (userType.isCustomer()) {
        // Important: Ensure that incrementAllIpCounter is called as the last argument as this call increments the
        // counter at the same time as returning whether the increment was successful or not.
        if (configHelper.isLanguageSupported(customerLanguage)
            && (!checkAndIncrementCounter || checkAndIncrementAllIpCounter())) {
            return RoutingDecision.ACTIVATION_ALL_IP;
        } else {
            return RoutingDecision.ACTIVATION_TDM_CUSTOMER;
        }
    } else {
        return RoutingDecision.ACTIVATION_TDM_CUSTOMER;
    }
}

@Override
public boolean isAllIpAdoptionPossible(final String mainPhoneNumber, final UserType userType,
    final String brokerId, boolean checkAndIncrementCounter, final String customerLanguage) {
    if (configHelper.isUseAllIp()) {
        if (!configHelper.isLanguageSupported(customerLanguage)) {
            return false;
        }
        if (userType.isShop()) {
            // Important: Ensure that incrementAllIpCounter is called as the last argument as this call increments
            // the counter at the same time as returning whether the increment was successful or not.
            return isShopOnWhitelist(brokerId) && isAllIpAdoptionTechnicallyPossible(mainPhoneNumber)
                && (!checkAndIncrementCounter || checkAndIncrementAllIpCounter());
        } else {
            if (configHelper.isUseAllIpWhitelist()) {
                return userType.isUserWhitelisted(brokerId);
            } else {
                // Important: Ensure that incrementAllIpCounter is called as the last argument as this call
                // increments the counter at the same time as returning whether the increment was successful or not.
                return userType.isUseAllIp() && isAllIpAdoptionTechnicallyPossible(mainPhoneNumber)
                    && (!checkAndIncrementCounter || checkAndIncrementAllIpCounter());
            }
        }
    }
    return false;
}
```

Nur ein Auszug der Implementation

# Ein Weg zu Entscheidungstabellen

## Die Konsequenzen

- Die Kommunikation zwischen Business Analyst und Entwickler gestaltet sich zunehmend schwieriger.
- Für den Entwickler wird die Lücke zwischen Anforderung und Implementation immer grösser.
- Die Business Analysten haben Mühe, neue Anforderungen in die bestehende Spezifikation zu integrieren.
- Die Entwickler haben Mühe, die Implementation weiterzuentwickeln.
- Die Tester haben Mühe, die Business Logik in der Spezifikation zu verstehen.

# Beispiel Spitalbesuch

## Die Regeln in Textform

Bereichsleiter Schmid möchte eine Mitarbeiterin im Krankenhaus besuchen. Er informiert sich telefonisch an der Information über die Besuchsmöglichkeiten und erhält folgende Antwort:

Die Patientin kann ohne Einschränkungen besucht werden, sofern keine ansteckende Krankheit vorliegt. Sonst werden Besuche ganz abgelehnt.



# Beispiel Spitalbesuch

## Implementation der Regeln in Code

```
/* Die Patientin kann ohne Einschränkungen besucht werden, sofern  
keine ansteckende Krankheit vorliegt. Sonst werden Besuche ganz  
abgelehnt. */
```

```
if (patientin.hatAnsteckendeKrankheit() {  
    besuch.ablehnen();  
}  
else {  
    besuch.erlauben();  
}
```

# Beispiel Spitalbesuch

## Die Regeln in Textform

Bereichsleiter Schmid möchte eine Mitarbeiterin im Krankenhaus besuchen. Er informiert sich telefonisch an der Information über die Besuchsmöglichkeiten und erhält folgende Antwort:

Die Patientin kann ohne Einschränkungen innerhalb der Besuchszeit besucht werden, sofern keine ansteckende Krankheit vorliegt. Außerhalb der Besuchszeit ist eine Schwester als Begleitung erforderlich. Falls die Patientin eine ansteckende Krankheit hat, werden Besuche ganz abgelehnt.

# Beispiel Spitalbesuch

## Implementation der Regeln in Code

```
/* Besuchszeit besucht werden, sofern keine ansteckende  
Krankheit vorliegt. Außerhalb der Besuchszeit ist eine  
Schwester als Begleitung erforderlich. Falls die Patientin  
eine ansteckende Krankheit hat, werden Besuche ganz  
abgelehnt. */
```

```
if (patientin.hatAnsteckendeKrankheit()) {  
    besuch.ablehnen();  
}  
else {  
    if (besuch.innerhalbBesuchszeit()) {  
        besuch.erlauben();  
    }  
    else {  
        besuch.erlaubenInBegleitungSchwester();  
    }  
}
```

# Beispiel Spitalbesuch

## Die Regeln in Textform

Bereichsleiter Schmid möchte eine Mitarbeiterin im Krankenhaus besuchen. Er informiert sich telefonisch an der Information über die Besuchsmöglichkeiten und erhält folgende Antwort:

Die Patientin kann ohne Einschränkungen innerhalb der Besuchszeit besucht werden, sofern keine ansteckende Krankheit vorliegt und sie kein Fieber hat. Außerhalb der Besuchszeit ist in diesem Fall eine Schwester als Begleitung erforderlich. Falls die Patientin eine ansteckende Krankheit hat, werden Besuche ganz abgelehnt. Wenn die Krankheit nicht ansteckend ist, die Patientin aber Fieber hat, darf der Besuch innerhalb der Besuchszeit maximal 30 Minuten betragen, außerhalb der Besuchszeit dürfen Patienten mit Fieber nicht besucht werden.

# Beispiel Spitalbesuch

## Implementation der Regeln in Code

```
if (patientin.hatAnsteckendeKrankheit() {  
    besuch.ablehnen();  
}  
else /* Patientin hat keine ansteckende Krankheit */ {  
    if (patientin.hatFieber()) {  
        if (besuch.innerhalbBesuchszeit()) {  
            besuch.erlaubenMaximal30Minuten();  
        }  
        else {  
            besuch.ablehnen();  
        }  
    }  
    else /* Patientin hat kein Fieber */ {  
        if (besuch.innerhalbBesuchszeit()) {  
            besuch.erlaubenNormal();  
        }  
        else {  
            besuch.erlaubenInBegleitungSchwester();  
        }  
    }  
}  
}
```



# Klassische Strukturierung einer Entscheidungstabelle

Entscheidungstabelle		Regeln								
		R1	R2	R3	R4	R5	R6	R7	R8	
Bedingungen	Bedingung 1	j	j	j	j	n	n	n	n	
	Bedingung 2	j	j	n	n	j	j	n	n	
	Bedingung 3	j	n	j	n	j	n	j	n	
Aktionen	Aktion 1	-	-	-	-	x	x	-	-	
	Aktion 2	-	-	-	-	x	x	-	-	
	Aktion 3	-	-	-	-	-	-	x	x	
	Aktion 4	-	x	-	x	-	x	-	-	
	Aktion 5	x	x	x	x	x	x	-	-	

# Beispiel Spitalbesuch

## Regeln als Entscheidungstabelle

Besuchsmöglichkeiten		Regeln							
		1	2	3	4	5	6	7	8
Bedingung	Ansteckende Krankheit (j/n)	j	j	j	j	n	n	n	n
	Besuch innerhalb Besuchszeit (j/n)	j	j	n	n	j	j	n	n
	Patientin hat Fieber (j/n)	j	n	j	n	j	n	j	n
Aktionen	Besuchszeit maximal 30 Minuten					x			
	Besuch ablehnen	x	x	x	x			x	
	Besuch mit Begleitung einer Schwester								x
	Normalbesuch in Besuchszeit						x		

# Beispiel Spitalbesuch

## Auswertung Entscheidungstabelle

1.

Input	Besuchsmöglichkeiten		Regeln							
			1	2	3	4	5	6	7	8
nein ja nein	Bedingung	Ansteckende Krankheit (j/n)	j	j	j	j	n	n	n	n
		Besuch innerhalb Besuchszeit (j/n)	j	j	n	n	j	j	n	n
		Patientin hat Fieber (j/n)	j	n	j	n	j	n	j	n
	Aktionen	Besuchszeit maximal 30 Minuten					x			
		Besuch ablehnen	x	x	x	x			x	
		Besuch mit Begleitung einer Schwester								x
		Normalbesuch in Besuchszeit						x		



# Beispiel Spitalbesuch

## Auswertung Entscheidungstabelle

1.

Input  
nein  
ja  
nein

Besuchsmöglichkeiten		Regeln							
		1	2	3	4	5	6	7	8
Bedingung	Ansteckende Krankheit (j/n)	j	j	j	j	n	n	n	n
	Besuch innerhalb Besuchszeit (j/n)	j	j	n	n	j	j	n	n
	Patientin hat Fieber (j/n)	j	n	j	n	j	n	j	n
Aktionen	Besuchszeit maximal 30 Minuten					x			
	Besuch ablehnen	x	x	x	x			x	
	Besuch mit Begleitung einer Schwester								x
	Normalbesuch in Besuchszeit						x		

Suche nach Übereinstimmung

2.

# Beispiel Spitalbesuch

## Auswertung Entscheidungstabelle

1.

Input  
nein  
ja  
nein

Besuchsmöglichkeiten		Regeln							
		1	2	3	4	5	6	7	8
Bedingung	Ansteckende Krankheit (j/n)	j	j	j	j	n	n	n	n
	Besuch innerhalb Besuchszeit (j/n)	j	j	n	n	j	j	n	n
	Patientin hat Fieber (j/n)	j	n	j	n	j	n	j	n
Aktionen	Besuchszeit maximal 30 Minuten					x			
	Besuch ablehnen	x	x	x	x			x	
	Besuch mit Begleitung einer Schwester								x
	Normalbesuch in Besuchszeit						x		

Suche nach Übereinstimmung

2.

Output: Aktion „Normalbesuch“

3.

# Beispiel Spitalbesuch: Auswertung in Java mit Rule Engine Drools

```
/* Excel einlesen */
StatelessKnowledgeSession session =
    factory.getStatelessKnowledgeSession();

/* Vorbereitung Output */
SpitalBesuchResultat besuch =
    new SpitalBesuchResultat();
session.setGlobal("SpitalBesuchResultat", session);

/* 1. Aufbereitung Input */
InputParameters inputParameters =
    inputParametersCreator.create("nein", "ja", "nein");

/* 2. Auswertung */
session.execute(Arrays.asList(inputParameters));

/* 3. Output: Aktion in besuch gespeichert */
```

# Beispiel Spitalbesuch: Vorteile der Auswertung mit einer Rule Engine

Wenn eine Entscheidungstabelle direkt im Code ausgewertet werden kann, bedeutet das eine klare Aufteilung der Verantwortlichkeiten

- **Verantwortung der Entwickler**
  - Input für die Tabelle korrekt aufbereiten
  - Output der Tabelle korrekt auswerten
  - nicht mehr für die Umsetzung der Logik verantwortlich
- **Verantwortung der Business Analysten**
  - Korrektheit der Regeln (vollständig und widerspruchsfrei)
- **Verantwortung der Tester**
  - Überprüfung, dass Verarbeitungskette  
Input => Auswertung => Output korrekt funktioniert
  - es ist aber nicht mehr nötig, jede Regel einzeln zu überprüfen

# Beispiel Spitalbesuch: Die Regeln als Entscheidungstabelle

Besuchsmöglichkeiten		Regeln							
Bedingung		1	2	3	4	5	6	7	8
	Ansteckende Krankheit (j/n)	j	j	j	j	n	n	n	n
	Besuch innerhalb Besuchszeit (j/n)	j	j	n	n	j	j	n	n
	Patientin hat Fieber (j/n)	j	n	j	n	j	n	j	n
Aktionen									
	Besuchszeit maximal 30 Minuten					x			
	Besuch ablehnen	x	x	x	x			x	
	Besuch mit Begleitung einer Schwester								x
Aktionen									
	Normalbesuch in Besuchszeit						x		

# Beispiel Spitalbesuch

## Konsolidierte Entscheidungstabelle

Besuchsmöglichkeiten		Regeln				
		1	5	6	7	8
<i>Anzahl abgedeckter Fälle</i>		4	1	1	1	1
Bedingung	Ansteckende Krankheit (j/n)	j	n	n	n	n
	Besuch innerhalb Besuchszeit (j/n)	*	j	j	n	n
	Patientin hat Fieber (j/n)	*	j	n	j	n
Aktionen	Besuchszeit maximal 30 Minuten		x			
	Besuch ablehnen	x			x	
	Besuch mit Begleitung einer Schwester					x
	Normalbesuch in Besuchszeit			x		

→  
Tabelle wächst horizontal mit Anzahl Regeln

# Beispiel Spitalbesuch

## Transponierte Entscheidungstabelle

Tabelle wächst vertikal mit Anzahl Regeln

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen			Aktionen			
			un- terdeckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normal- besuch in Besuchszeit
Regeln	1	4	j	*	*		x		
	5	1	n	j	j	x			
	6	1	n	j	n				x
	7	1	n	n	j		x		
	8	1	n	n	n			x	

Diese Entscheidungstabelle ist **vollständig**: Alle 8 möglichen Fälle sind abgedeckt.

Diese Entscheidungstabelle ist **eindeutig**: Die Bedingungen schliessen sich gegenseitig aus.

# Beispiel Spitalbesuch

## Transponierte Entscheidungstabelle

Tabelle wächst vertikal mit Anzahl Regeln

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen			Aktionen			
			Ansteckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normalbesuch in Besuchszeit
Regeln	1	4	j	*	*		x		
	5	1	n	j	j	x			
	6	1	n	j	*				x
	7	1	n	n	j		x		
	8	1	n	n	n			x	

Diese Entscheidungstabelle ist **mehrdeutig**: Die Bedingungen der Regeln 5 und 6 schliessen sich nicht gegenseitig aus.

Die Bedingungswerte n/j/j werden durch beide Regeln abgedeckt, führen aber zu *unterschiedlichen* Aktionen.



# Entscheidungstabellen

## Vorteile grundsätzlicher Natur

- Entscheidungstabellen kommunizieren klar die fachliche Logik: Die tabellarische Darstellung der Regeln stellt Bedingungen und Aktionen kompakt und übersichtlich dar.
- Die Struktur der Tabelle ist präzise formal definiert, insbesondere auch die zulässigen Wertebereiche aller Bedingungen. Jede Entscheidungstabelle könnte auf Wahrheitstabellen reduziert werden, die in der theoretischen Informatik eine wichtige Rolle spielen.
- Dadurch ist eine ausgefüllte Tabelle überprüfbar:
  - Ist die Tabelle vollständig?
  - Gibt es Mehrdeutigkeiten in der Tabelle?
  - Gibt es Widersprüche in der Tabelle?

# Entscheidungstabellen

## Vorteile in der Anwendung

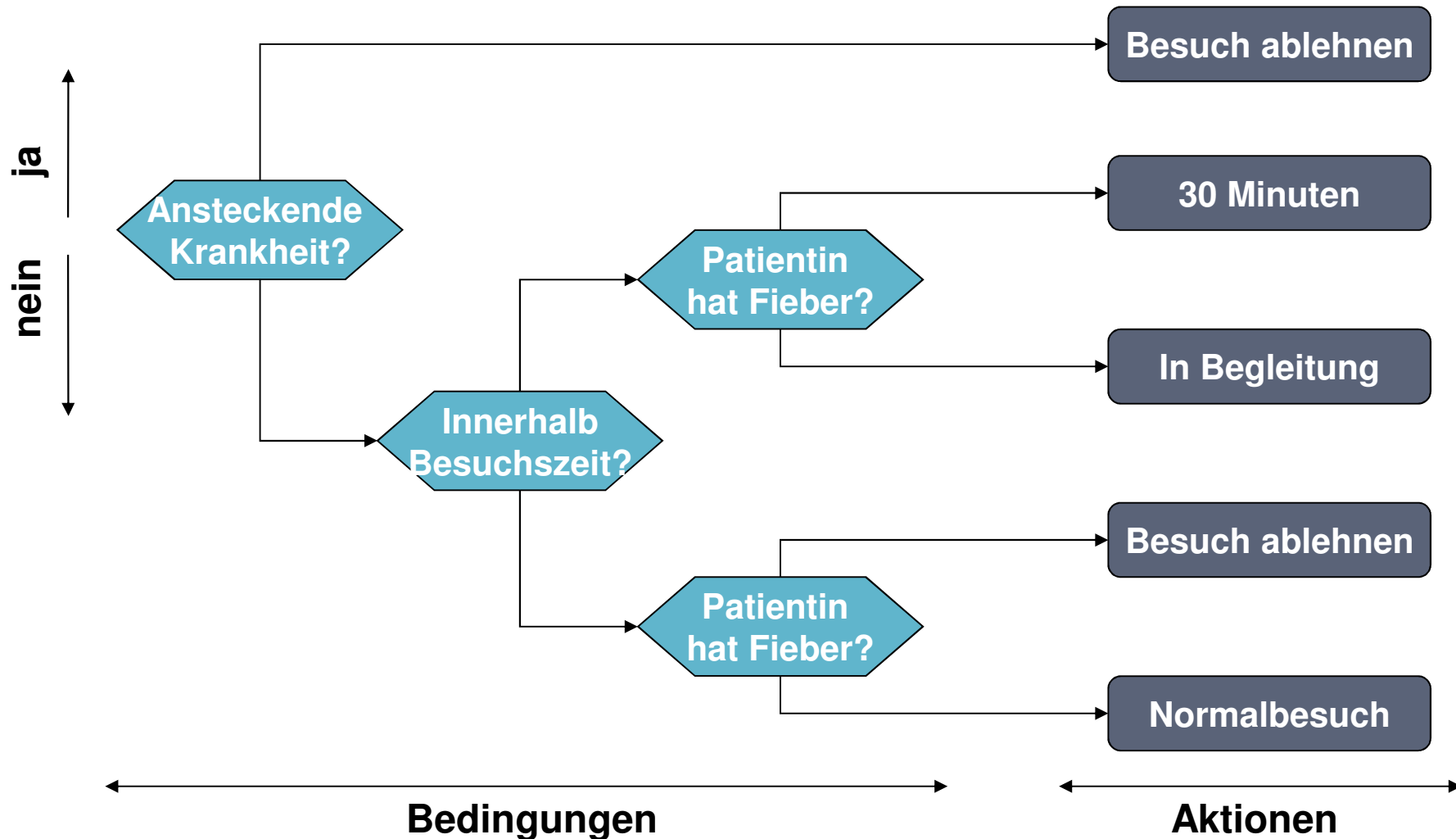
- Mit dem Einsatz von Entscheidungstabellen übernehmen Business Analysten die nicht die Verantwortung für die Spezifikation, sondern gleichzeitig auch für einen grossen Teil der Implementation.
- Entscheidungstabellen sind einfach wartbar: Eine Aktion kann jederzeit einfach angepasst werden, sogar ohne Code-Anpassung.
- Entscheidungstabellen sind erweiterbar:
  - Neue Bedingungen können hinzugefügt werden. Die bestehenden Regeln müssen alle angepasst werden.
  - Regeln bzw. Bedingungswerte einer Regel können angepasst werden. Es muss sichergestellt werden, dass Tabelle weiterhin vollständig und widerspruchsfrei ist.
- Entscheidungstabellen können auch grafisch als Entscheidungsbäume dargestellt werden.

# Entscheidungstabellen

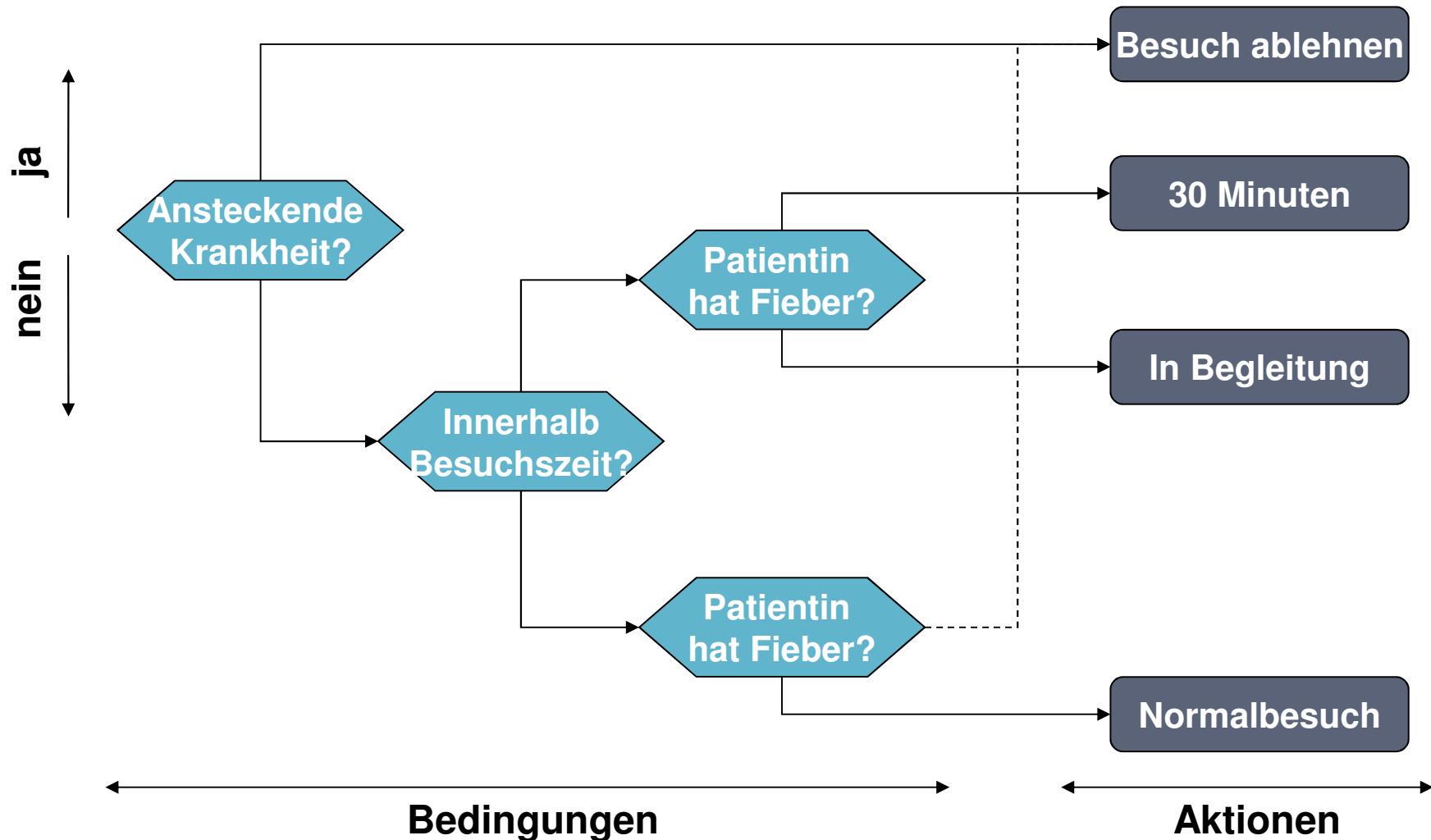
## Nachteile in der Anwendung

- Verwendung einer Rule Engine: Je nach verwendetem Tool muss der nicht unerhebliche Speicherbedarf der Rule Engine berücksichtigt werden.
- Wartbarkeit: Auch wenn Entscheidungstabellen prinzipiell gut wartbar sind, gibt es ein Caveat: Die Unterschiede zwischen zwei Versionen sind unter Umständen – ja nach Tool – schwierig nachträglich zu ermitteln.

# Beispiel Spitalbesuch: Darstellung als Entscheidungsbaum

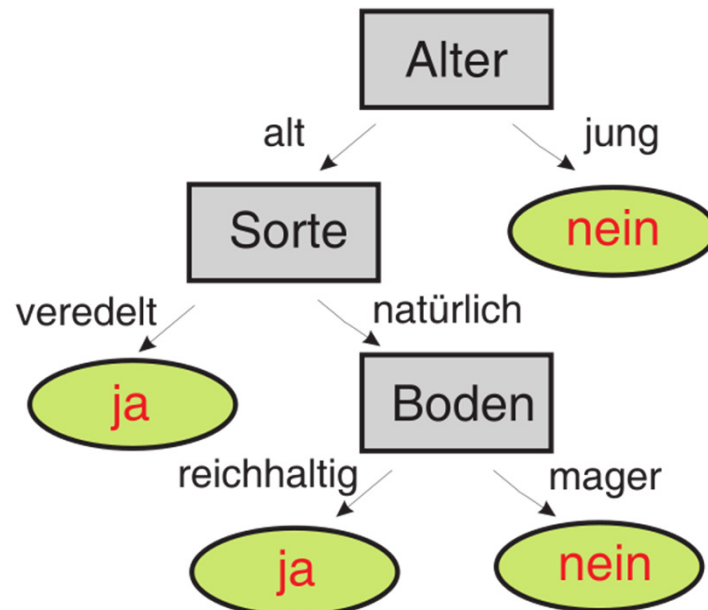


# Beispiel Spitalbesuch: Darstellung als Entscheidungsgraph



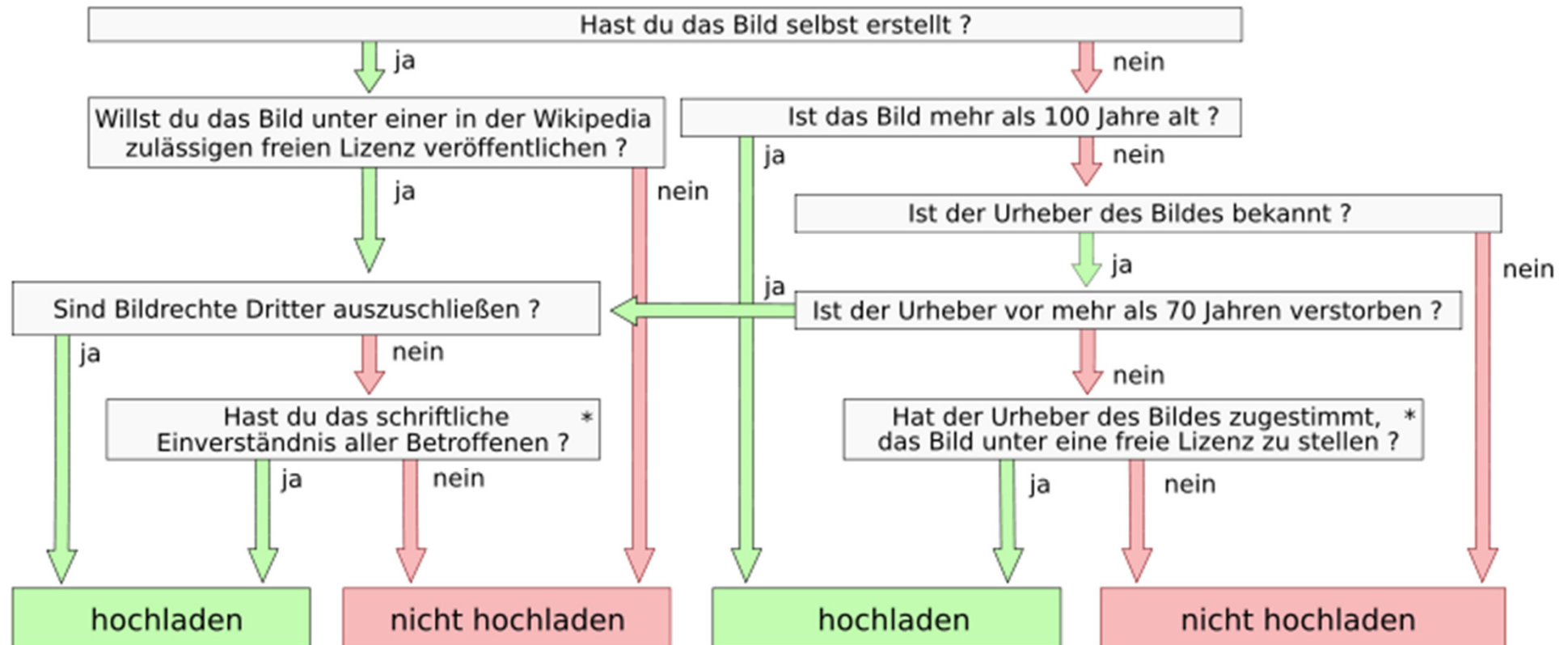
# Darstellungsform: Entscheidungsbaum oder Entscheidungstabelle?

Prognose Früchtetragen		Anzahl Fälle	Bedingungen			Aktionen
			Alter	Sorte	Boden	
Regeln	1	4	jung	*	*	nein
	2	2	alt	veredelt	*	ja
	3	1	alt	natürlich	reichhaltig	ja
	4	1	alt	natürlich	mager	nein



Da die Bedingungs-  
werte eine „Dreiecks-  
struktur“ haben, lassen  
sich die Regeln sehr  
übersichtlich als  
Entscheidungsbaum  
darstellen.

# Beispiel von Wikipedia: Darstellung als Entscheidungsbaum



\* Sämtliche Anfragen an den Rechteinhaber sind an das OTRS ([permission-de@wikimedia.org](mailto:permission-de@wikimedia.org)) weiterzuleiten.

# Beispiel von Wikipedia: Darstellung als Entscheidungstabelle

Bilder hochladen?		Anzahl abgedeckter Fälle	Bedingungen								Aktionen
			Bild selber erstellt?	Unter freier Lizenz veröffentlichen?	Bildrechte Dritter ausgeschlossen?	Schriftl. Einverständnis aller?	Bild mehr als 100 Jahre alt?	Urheber des Bildes bekannt?	Urheber vor mehr als 70 Jahren verstorben?	Urheber mit freier Lizenz einverstanden?	
Regeln	1	32	ja	ja	ja	*	*	*	*	*	hochladen
	2	16	ja	ja	nein	ja	*	*	*	*	hochladen
	3	16	ja	ja	nein	nein	*	*	*	*	nicht hochladen
	4	64	ja	nein	*	*	*	*	*	*	nicht hochladen
	5	64	nein	*	*	*	ja	*	*	*	hochladen
	6	8	nein	*	ja	*	nein	ja	ja	*	hochladen
	7	4	nein	*	nein	ja	nein	ja	ja	*	hochladen
	8	4	nein	*	nein	nein	nein	ja	ja	*	nicht hochladen
	9	8	nein	*	*	*	nein	ja	nein	ja	hochladen
	10	8	nein	*	*	*	nein	ja	nein	nein	nicht hochladen
	11	32	nein	*	*	*	nein	nein	*	*	nicht hochladen
		256									

Nicht mehr sehr übersichtlich.

Idee hinter den Regeln schwer ersichtlich.



# Beispiel von Wikipedia: Als Entscheidungstabelle mit „else“

Bilder hochladen?		Anzahl abgedeckter Fälle	Bedingungen								Aktionen
			Bild selber erstellt?	Unter freier Lizenz veröffentlichen?	Bildrechte Dritter ausgeschlossen?	Schriftl. Einverständnis aller?	Bild mehr als 100 Jahre alt?	Urheber des Bildes bekannt?	Urheber vor mehr als 70 Jahren verstorben?	Urheber mit freier Lizenz einverstanden?	
Regeln	1	32	ja	ja	ja	*	*	*	*	*	hochladen
	2	16	ja	ja	nein	ja	*	*	*	*	hochladen
	3	64	nein	*	*	*	ja	*	*	*	hochladen
	4	8	nein	*	ja	*	nein	ja	ja	*	hochladen
	5	64	nein	*	nein	ja	nein	ja	ja	*	hochladen
	6	8	nein	*	*	*	nein	ja	nein	ja	hochladen
	7	64	else								nicht hochladen
		256									

Auflistung aller positiven Fälle.

Idee hinter den Regeln besser ersichtlich.

**Kann nicht mehr auf Vollständigkeit geprüft werden!**

# Beispiel von Wikipedia: Als Regeln/ Formeln formulieren

**6 Regeln der folgenden Art (für jeden Fall „hochladen“ eine Regel):**

(Bild selber erstellt) UND (Unter freier Lizenz veröffentlichen) UND  
(Bildrechte Dritter ausgeschlossen)

ODER

(Bild selber erstellt) UND (Unter freier Lizenz veröffentlichen) UND  
NICHT (Bildrechte Dritter ausgeschlossen) UND (Schriftl.  
Einverständnis aller)

ODER

...

**Gut geeignet, wenn es nur wenige  
positive (oder negative) Fälle gibt**

# Inhalt

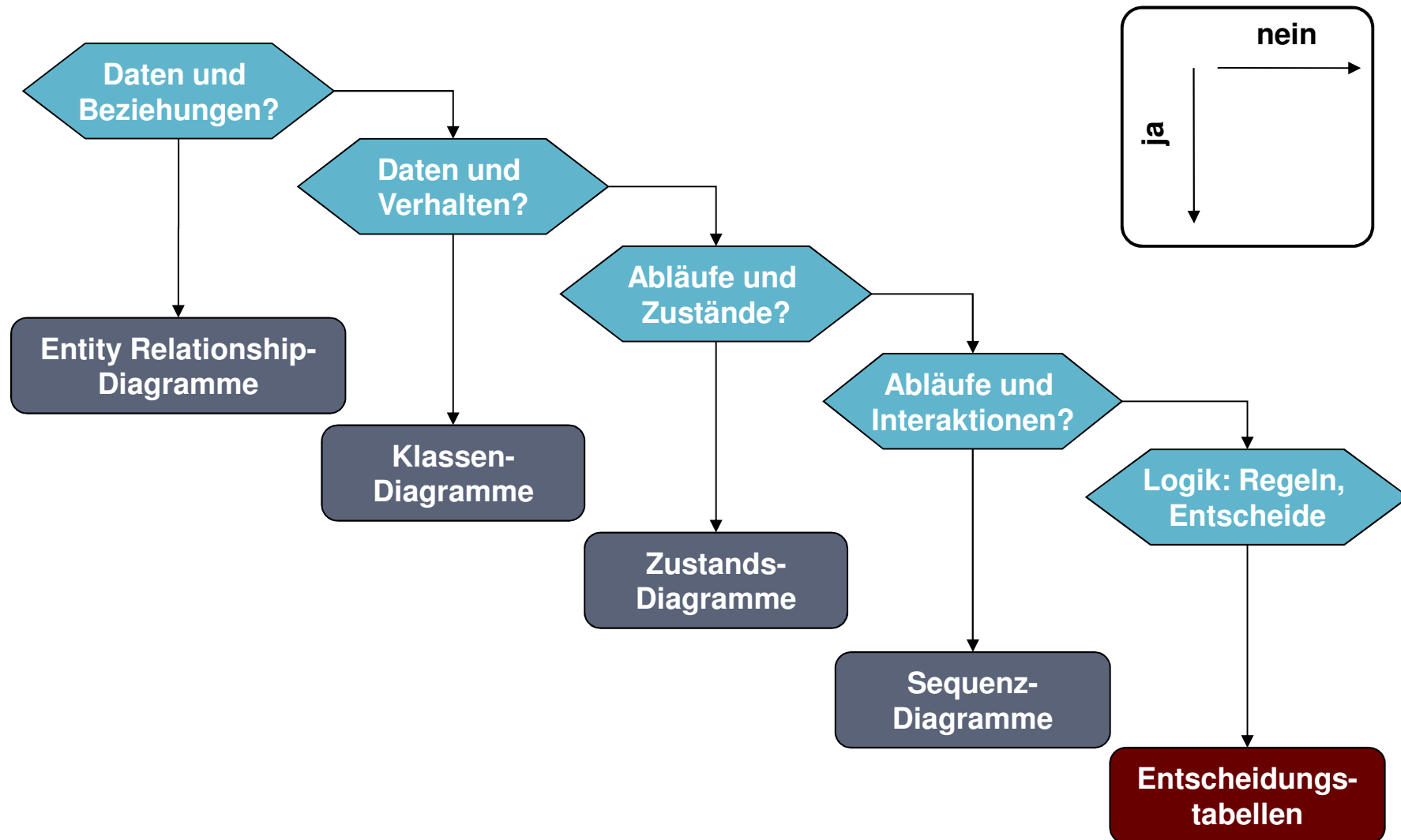
Motivation

Anwendungsbereich

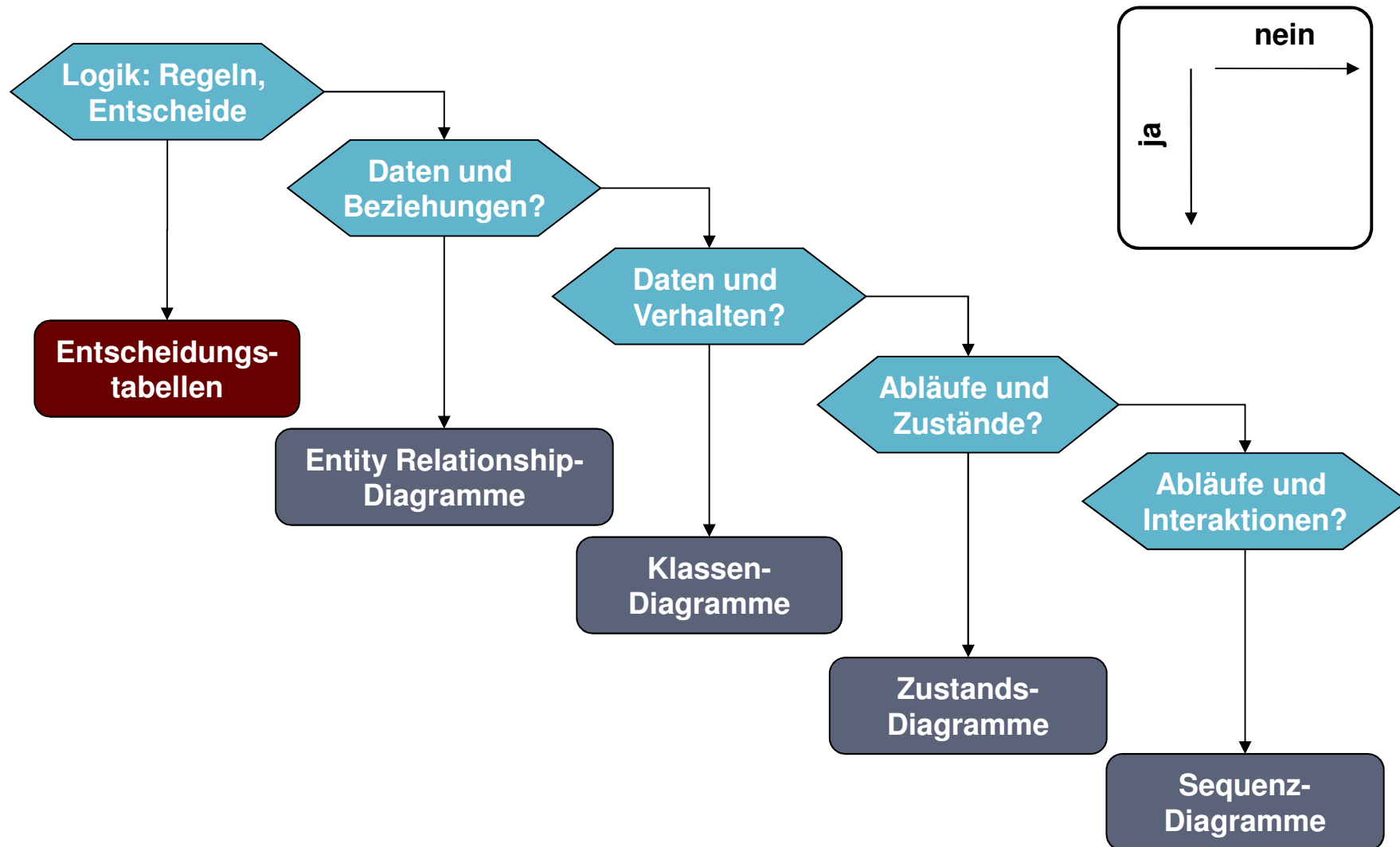
Modellierung

Beispiele

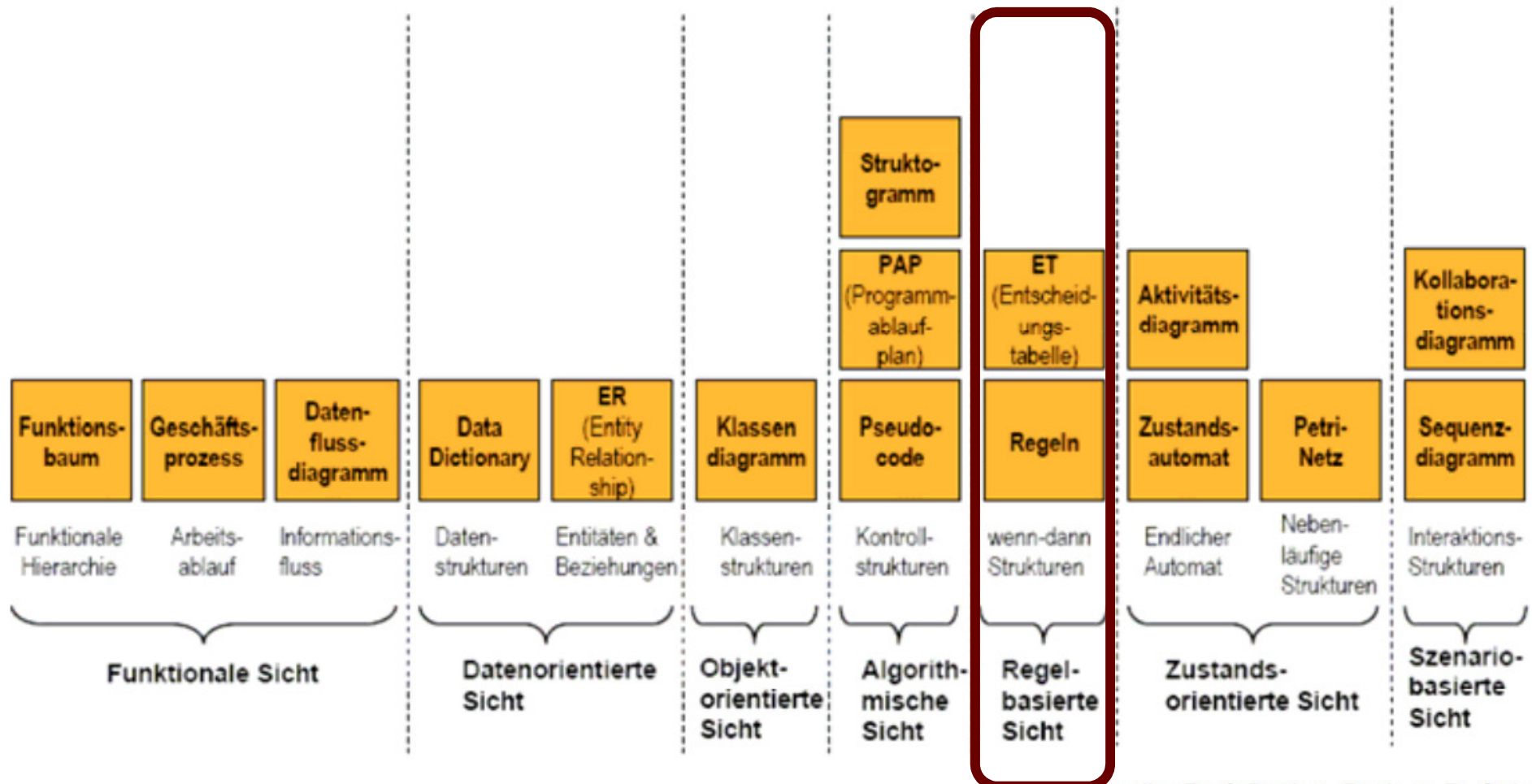
# Anwendungsbereich: Wann Entscheidungstabellen einsetzen?



# Anwendungsbereich: Wann Entscheidungstabellen einsetzen?



# Einordnung Entscheidungstabellen in Methoden zur Systemmodellierung



# Inhalt

Motivation

Anwendungsbereich

Modellierung

Beispiele

# Beispiel Spitalbesuch vollständig und eindeutig

Besuchsmöglichkeiten		Anzahl abge- deckter Fälle	Bedingungen			Aktionen			
			an- steckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normal- besuch in Besuchszeit
Regeln	1	4	j	*	*		x		
	5	1	n	j	j	x			
	6	1	n	j	n				x
	7	1	n	n	j		x		
	8	1	n	n	n			x	

Diese Entscheidungstabelle ist **vollständig**: Alle 8 möglichen Fälle sind abgedeckt.

Diese Entscheidungstabelle ist **eindeutig**: Die Bedingungen schliessen sich gegenseitig aus.



# Beispiel Spitalbesuch

## Mehrfach definiert

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen			Aktionen			
			Ansteckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normalbesuch in Besuchszeit
Regeln	1	4	j	*	*		X		
	5	1	n	j	j	X			
	6	1	n	j	*	X			
	7	1	n	n	j		X		
	8	1	n	n	n			X	

Diese Entscheidungstabelle ist **mehrfach definiert**: Die Bedingungen der Regeln 5 und 6 schliessen sich nicht gegenseitig aus.

Die Bedingungswerte n/j/j werden durch beide Regeln abgedeckt und führen zur **gleichen** Aktion.

# Beispiel Spitalbesuch

## Mehrdeutig und widersprüchlich

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen			Aktionen			
			Ansteckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normalbesuch in Besuchszeit
Regeln	1	4	j	*	*		x		
	5	1	n	i	i	x			
	6	1	n	j	*				x
	7	1	n	n	j		x		
	8	1	n	n	n			x	

Diese Entscheidungstabelle ist **mehrdeutig** und **widersprüchlich**: Die Bedingungen der Regeln 5 und 6 schliessen sich nicht gegenseitig aus.

Die Bedingungswerte n/j/j werden durch beide Regeln abgedeckt, führen aber zu **unterschiedlichen** Aktionen.

# Regeln für die Definition des Inputs

## Reduce to the max:

- Keine unnötigen Variablenwerte
- Keine unnötigen Variablen
- Nur unabhängige Variablen
- Kombination von Subvariablen

# Minimaler Werteraum

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen			Aktionen			
			Ansteckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (kein, schwach, stark)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normal- besuch in Besuchszeit
Regeln	1	6	j	*	*		X		
	7	1	n	j	schwach	X			
	8	1	n	j	stark	X			
	9	1	n	j	kein				X
	10	1	n	n	schwach		X		
	11	1	n	n	stark		X		
	12	1	n	n	kein			X	

Der Werteraum der Variablen sollte minimal sein.  
Dadurch kann sichergestellt werden dass nicht Regeln  
entstehen welche redundant sind.

# Keine unnötigen Variablen

Besuchsmöglichkeiten		Anzahl abge- deckter Fälle	Bedingungen				Aktionen			
			An- steckende Krankheit (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besucher volljährig (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normal- besuch in Besuchszeit
Regeln	1	8	j	*	*	*		x		
	9	2	n	j	j	j	x			
	11	2	n	j	n	n				x
	13	2	n	n	j	j		x		
	15	2	n	n	n	n			x	
	17	2	n	j	j	n	x			
	19	2	n	j	n	j				x
	21	2	n	n	j	n		x		
	23	2	n	n	n	j			x	

Unnötige Variablen müssen nicht immer offensichtlich sein. Falls für eine Variable nur Sterne stehen, dann ist sie offensichtlich nicht notwendig. Es kann aber auch sein, dass die Entscheidungstabelle nicht minimiert ist und es deshalb nicht offensichtlich ist, dass es unnötige Variablen gibt.

# Unabhängige Variablen

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen				Aktionen				
			Ansteckende Krankheit (j/n)	Patient darf in die Cafeteria gehen (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normalbesuch in Besuchszeit	Nicht möglich
Regeln	1	4	j	n	*	*		x			
	5	4	j	j	*	*					x
	9	1	n	n	j	j	x				
	10	1	n	j	j	j					x
	11	1	n	j	j	n				x	
	12	1	n	n	j	n					x
	13	1	n	n	n	j		x			
	14	1	n	j	n	j					x
	15	1	n	j	n	n			x		
	16	1	n	n	n	n					x

Die Variable „Patient darf in die Cafeteria gehen“ ist abhängig davon, ob der Patient Fieber hat und ob er eine ansteckende Krankheit hat. Es kann aber durchaus sinnvoll argumentiert werden, dass Besuch davon abhängig ist, ob der Patient in die Cafeteria darf. Genauere Analyse der Regeln zeigt aber, dass in diesen Regeln die eigentlich entscheidenden Faktoren Fieber und die ansteckende Krankheit ist.

# Kombinierung von Subvariablen

Besuchsmöglichkeiten		Anzahl abgedeckter Fälle	Bedingungen					Aktionen			
			Tröpfcheninfektion (j/n)	Schmierinfektion (j/n)	Kontaktinfektion (j/n)	Besuch innerhalb Besuchszeit (j/n)	Patientin hat Fieber (j/n)	Besuchszeit maximal 30 Minuten	Besuch ablehnen	Besuch mit Begleitung einer Schwester	Normalbesuch in Besuchszeit
Regeln	1	16	j	*	*	*	*		x		
	17	16	*	j	*	*	*		x		
	33	16	*	*	j	*	*		x		
	49	1	n	n	n	j	j	x			
	50	1	n	n	n	j	n				x
	51	1	n	n	n	n	j		x		
	52	1	n	n	n	n	n			x	

Für die Entscheidung, ob der Patient Besucher empfangen darf oder nicht, spielt nicht die Art der Infektionsübertragung eine Rolle, sondern einzig, ob der Patient die Krankheit bei einem Besuch auf den Besucher übertragen kann. D.h. die konkrete Übertragungsart spielt bei der Entscheidung keine Rolle.

# Inhalt

Motivation

Anwendungsbereich

Modellierung

Beispiele



# Beispiel

<http://swt.cs.tu-berlin.de/lehre/mwsp/wso607/ausarbeitungen/Ausarbeitung-10.pdf>

[illegible]

# Beispiel

[https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/infll\\_sso6/inf\\_II\\_kapitel\\_05.pdf](https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/infll_sso6/inf_II_kapitel_05.pdf)

	1	2	3	4	5	6	7
Bestellbetrag > Kreditlimite	N	J	J	J	J	J	J
Bestellbetrag > 1.5 * Kreditlimite	–	N	N	J	J	J	J
Sonderkunde	–	J	N	J	N	N	N
Jahresumsatz > 50000	–	–	–	–	N	J	N
2-Monats-Umsatz > 20000	–	–	–	–	J	–	N
Bestellung ausliefern	X	X					
Bestellung an Verkaufsleiter			X	X	X	X	
Bestellung zurückweisen							X

# Beispiel

[https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/infil\\_sso6/inf\\_II\\_kapitel\\_05.pdf](https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/infil_sso6/inf_II_kapitel_05.pdf)

	1	2	3	4	5	6	7
Bestellbetrag > Kreditlimite	N	J	J	J	J	J	J
Bestellbetrag > 1.5 * Kreditlimite	–	N	N	J	J	J	J
Sonderkunde	–	J	N	J	N	N	N
Jahresumsatz > 50000	–	–	–	–	N	J	N
2-Monats-Umsatz > 20000	–	–	–	–	J	–	–
Bestellung ausliefern	X	X					
Bestellung an Verkaufsleiter			X	X	X	X	
Bestellung zurückweisen							X

Würde Spalte 2 gestrichen, wäre die Behandlung der Bestellung von Sonderkunden mit:  
 $\text{Limite} < \text{Bestellbetrag} \leq 1,5 * \text{Limite}$  nicht spezifiziert.

Egal statt „Nein“ führt zu Widerspruch

# Beispiel

[http://www.trautwein.fh-](http://www.trautwein.fh-aachen.de/Download/SWE/SWE_Ausarbeitungen_WS2010/A_Entscheidungstabellen_2010.pdf)

[aachen.de/Download/SWE/SWE\\_Ausarbeitungen\\_WS2010/A\\_Entscheidungstabellen\\_2010.pdf](http://www.trautwein.fh-aachen.de/Download/SWE/SWE_Ausarbeitungen_WS2010/A_Entscheidungstabellen_2010.pdf)

	Bedingungsteil	Textteil	Anzeigeteil								
			R1	R2	R3	R4		R10	R11	R12	
	Bedingungsteil	Wieviel Geld hab ich dabei [0, 20]	< 2	< 2	< 2	< 2	..	> 5	> 5	> 5	WENN-Teil
		Bin ich auf Diät [ja, nein]	n	n	j	j	..	n	j	j	
		Bin ich Vegetarier [ja, nein]	n	j	n	j	..	j	n	j	
	Aktionsteil										DANN-Teil
		frit. Fisch + Pommes (2,60)					..	X			
		Gemüse + Reis (1,80)	X	X	X	X	..	X	X	X	
		Rindersteak + Pilsssoße (4,00)					..				
		Tagessuppe (0,50)	X	X	X	X	..	X	X	X	
		Sushi (5,00)					..	X	X	X	

# Beispiel

<http://ti.uni-due.de/ti/de/education/teaching/ss10/pet/download/Vorlesung%202.pdf>

## 2. Allgemeine Modellbildung

### 2.2 Entscheidungstabellen

Kapitel 2: Allgemeine  
Modellbildung  
-Klassische Konzepte des  
SWE-

Erweiterte Entscheidungstabellen für das  
**Beispiel „Weihnachtsgeschenke“**

Weihnachts- geschenke	Regeln				
	1	2	3	4	5
Bestellwert	(0, 5.000)	(5.000, 20.000)	(5.000, 20.000)	> 20.000	> 20.000
Anti-Alkoholiker	-	J	N	J	N
Glückwunschkarte	X	X	X	X	X
Präsent		Buch	Wein	Buch	Wein
Persönlicher Anruf				X	X

# Beispiel

[http://en.wikipedia.org/wiki/Decision\\_table](http://en.wikipedia.org/wiki/Decision_table)

**Printer troubleshooter**

		Rules							
Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognised	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check the printer-computer cable	X		X					
	Ensure printer software is installed	X		X		X		X	
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				