8 First-Order-Logic (FOL)

- Sprache der Aussagenlogik zu schwach, um Wissen aus komplexen Umgebungen präzise zu repräsentieren
- ein großer Teil unseres Allgemeinwissens kann in FOL (Prädikatenlogik) repräsentiert werden
- Die formale Sprache der FOL ist bzgl. ihrer Syntax und Semantik gut erforscht
- Programmiersprachen für Prädikatenlogik

Prozedurale Sprachen

- haben keinen (eingebauten) Inferenz-Mechanismus, um Fakten aus anderen Fakten abzuleiten. Aktualisierungen einer Datenstruktur erfolgt durch Domain-spezifische Prozeduren. Die Details und das Wissen über Domains besitzt der Programmierer
- haben für jede Variable nur einen Wert; nicht einfach möglich: „*es befindet sich ein Falltür in [2,2] oder [3,1]*“ oder „*Wenn sich das Wumpus in [1,1] befindet, befindet es sich nicht in [2,2].*“
- erlauben zwar den Wert "*unbekannt*" für eine Variable, haben aber nicht die Ausdruckskraft, um partielle Informationen zu verarbeiten

Aussagenlogik - deklarative Sprache

Die **Aussagenlogik**

- ist eine deklarative Sprache, weil ihre Semantik auf einer Wahrheitsrelation zwischen Sätzen und möglichen Welten basiert
- kann mit Hilfe von Disjunktion und Negation auch mit partiellen Informationen umgehen
- ist **kompositional**, d.h. die Bedeutung eines Satzes ist eine Funktion der Bedeutung seiner Bestandteile
- kann nur umständlich eine Umgebung mit vielen Objekten beschreiben

Ziel: deklarative Semantik

- Deklarative, kompositionale Semantik, die kontextunabhängig und eindeutig ist
- Objekte, Relationen, Funktionen
  - Objekte: Menschen, Häuser, ...
  - Relationen:
    - unäre: rot, rund, falsch, weiblich, ...
    - n-äre: ist Bruder von, ist größer als, liegt innerhalb, ist Teil von, kommt zwischen, ...
  - Funktionen: ist Vater von, ist bester Freund, ist Dritter von rechts, ist einer mehr als, liegt am Anfang von, ...

Eigenschaft

In der Aussagenlogik und in der FOL repräsentiert ein Satz ein Fakt, und der Agent glaubt entweder,

- dass der Satz wahr ist,
- dass der Satz falsch ist, oder
- er hat keine Meinung

Formale Sprachen und ihre Bindungen

Sprache	Ontologische Bindung (Was in der Welt existiert)	Epistemologische Bindung (Was ein Agent im Hinblick auf Fakten glaubt)
Aussagenlogik	Fakten	wahr/falsch/unbekannt
First-Order-Logik	Fakten, Objekte, Relationen	wahr/falsch/unbekannt
Temporale Logik	Fakten, Objekte, Relationen, Zeiten	wahr/falsch/unbekannt
Wahrschein-lichkeitstheorie	Fakten	Glaubensgrad $\in [0,1]$
Fuzzy Logik	Fakten mit einem Wahrscheinlichkeitsgrad $\in [0,1]$	bekannter Intervallwert

Modelle für die FOL

Modelle einer logischen Sprache sind formale Strukturen, die die möglichen Welten bilden.

Beispiel – König-Modell mit 5 Objekten

brother(Richard Löwenherz, John)

on head(crown, John)

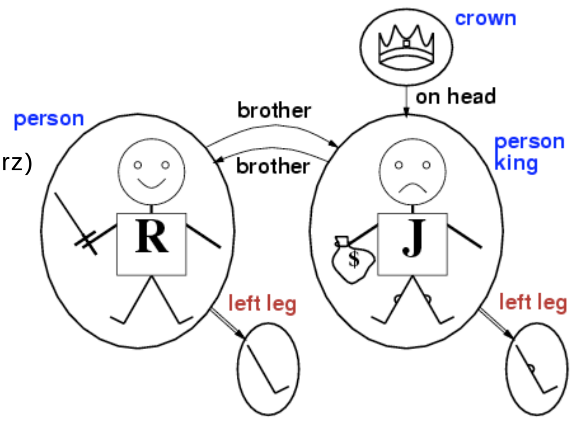
is person(John)

is king(John)

has crown(John)

left leg(Richard Löwenherz)

left leg(John)



Symbole und Interpretationen

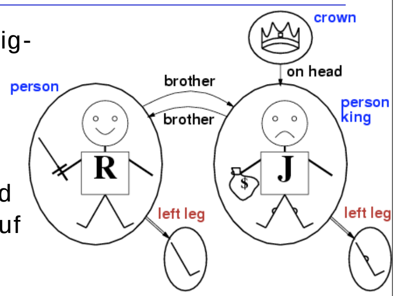
Die grundlegenden syntaktischen Elemente der FOL sind die Symbole, die für Objekte, Relationen und Funktionen stehen. Es gibt also 3 Arten von Symbolen:

- **Konstantensymbole** für Objekte
- **Prädikatsymbole** für Relationen (mit Stelligkeit)
- **Funktionssymbole** für Funktionen (mit Stelligkeit)

Die Semantik muss Sätze mit Modellen in Beziehung bringen, um den Wahrheitsgehalt festzulegen. Dazu wird eine **Interpretation** gebraucht, die genau angibt, welche Objekte, Relationen und Funktionen mit welchen Symbolen angesprochen werden.

Beabsichtigte Interpretation des König-Modells

Eine mögliche Interpretation für das König-Modell – die als **beabsichtigte Interpretation** bezeichnet wird – ist folgende:



- *Richard* bzw. *R* bezieht sich auf Richard Löwenherz; *John* bzw. *J* bezieht sich auf König John
- *Bruder* bzw. *brother* bezieht sich auf die Bruderbeziehung; *AufDemKopf* bzw. *on head* bezieht sich auf die Relation „auf dem Kopf“, die zwischen Krone und König John gilt; *person*, *king* und *crown* beziehen sich auf die Objektmengen, die Personen, Könige und Kronen sind.
- *left leg* bezieht sich auf die Funktion „linkes Bein“

Wahrheit eines Satzes

Die Wahrheit eines Satzes wird durch ein Modell und eine Interpretation für die Symbole des Satzes festgelegt.

Darum sind logische Konsequenz, Gültigkeit usw. im Hinblick auf *alle möglichen Modelle* und *alle möglichen Interpretationen* definiert.

Beachte, dass die Anzahl der **Domänenelemente** in jedem Modell **unbegrenzt** sein kann, z.B. können Domänenelemente ganze Zahlen oder reelle Zahlen sein. Damit ist auch die Anzahl möglicher Modelle unbegrenzt ebenso wie die Anzahl der Interpretationen.

Eine Überprüfung der logischen Konsequenz durch Auflistung aller möglichen Modelle, was in der Aussagenlogik möglich war, ist bei der FOL nicht mehr möglich.

Mit den Symbolen aus dem König-Modell gibt es etwa 10^{25} Kombinationen für eine Domäne mit 5 Objekten.

Syntax of FOL

Sentence	→	Atomic Sentence
	|	(Sentence Connective Sentence)
	|	Quantifier Variable, … Sentence
	|	¬ Sentence
Atomic Sentence	→	Predicate(Term, …) | Term = Term
Term	→	Function(Term, …)
	|	Constant
	|	Variable

Constants	KingJohn, 2, ...
Predicates	Brother, >,...
Functions	Mother, LeftLeg,...
Variables	x, y, a, b,...
Connectives	$\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
Equality	=
Quantifiers	$\forall$, $\exists$

Terme

Term: logischer Ausdruck, der sich auf ein Objekt bezieht, z.B. $f(t_1, \ldots,)$. *LinkesBein(John)*

Konstantensymbole sind Terme. Es ist aber unpraktisch, jedem Objekt einen Namen zu geben, d.h. für jedes Objekt ein separates Symbol zu verwenden.

Z.B. geben wir dem linken Bein von König John keinen Namen, sondern sagen „König Johns linkes Bein".

Genau dafür verwendet man Funktionssymbole wie *LinkesBein(John)*.

Komplexer Term: Funktionssymbol mit einer in Klammern eingeschlossenen Liste mit Termen als Argumente

Komplizierte Art von Name.

Kein Funktionsaufruf mit Rückgabewert!

Semantik von Termen

Betrachte den Term $f(t_1, \ldots, t_n)$.

Das Symbol f verweist auf irgendeine Funktion im Modell, die Argumentterme $t_1, \ldots, t_n$ verweisen auf Objekte in der Domäne.

Der Term als Ganzes verweist auf das Objekt „linkes Bein von König John".

Atomare Sätze

Atomarer Satz: drückt ein **Fakt** aus.

Beispiele:

- Bruder(Richard, John)
- Verheiratet(Vater(Richard), Mutter(John))

Ein atomarer Satz ist in einem bestimmten Modell unter einer bestimmten Interpretation ***wahr****, wenn die durch das Prädikatsymbol angegebene Relation für die von den Argumenten angegebenen Objekte gilt.*

Komplexe Sätze

Durch logische Verknüpfungen werden komplexere Sätze konstruiert

Beispiele:

¬Bruder(LinkesBein(Richard), John)

Bruder(Richard,John) ∧ Bruder(John,Richard)

König(Richard) ∨ König(John)

¬König(Richard) ⇒ König(John)

Quantoren in FOL

Die FOL enthält zwei Standardquantoren:

∀ - universell

∃ - existentiell.

Mit dem **Allquantor ∀** können Eigenschaften ganzer Objektsammlungen ausgedrückt werden. Die Objekte müssen nicht mit ihrem Namen aufgelistet werden. Der Allquantor trifft somit Aussagen über jedes Objekt.

Mit dem **Existenzquantor ∃** kann eine Aussage zu einem bestimmten Objekt im Universum gemacht werden, ohne es per Namen zu nennen.

Allquantor ∀

Alltägliche Aussagen wie „Quadrate, die an das Quadrat mit dem Wumpus grenzen, stinken“ oder „Alle Könige sind Personen“ können in der FOL mit Hilfe des Allquantors einfach ausgedrückt werden.

Beispiel: $\forall$ x König(x) $\Rightarrow$ Person(x)

Sprechweise: Für alle x gilt, wenn x ein König ist, dann ist x auch eine Person, oder kurz: alle Könige sind Personen

Das Symbol x wird als **Variable** bezeichnet.

Intuitiv besagt der Satz $\forall$ x P, wobei P ein beliebiger logischer Ausdruck ist, dass P für jedes Objekt x wahr ist.

Genauer: $\forall$ x P ist in einem bestimmten Modell unter einer bestimmten Interpretation wahr, wenn P in allen möglichen erweiterten Interpretationen, die aus der gegebenen Interpretation konstruiert werden können, wahr ist.

Beispiel: universelle Quantifizierung im König-Modell

Die Interpretation des Modells kann auf mehrere Arten erweitert werden:

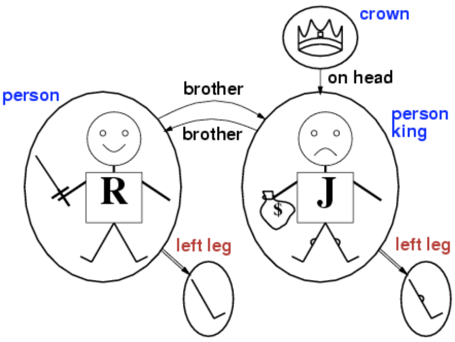
x → Richard Löwenherz

x → König John

x → Richards linkes Bein

x → Johns linkes Bein

x → die Krone



Der allquantifizierte Satz $\forall$ x König(x) $\Rightarrow$ Person(x) ist unter der ursprünglichen Interpretation wahr, wenn der Satz König(x) $\Rightarrow$ Person(x) in jeder der 5 erweiterten Interpretationen wahr ist.

Beispiel – universelle ...

Mit der erweiterten Interpretation des Modells ist der universell quantifizierte Satz

$\forall$x König(x) $\Rightarrow$ Person(x)

äquivalent mit der Zusicherung der folgenden 5 Sätze:

Richard Löwenherz ist ein König $\Rightarrow$ Richard Löwenherz ist eine Person

König John ist ein König $\Rightarrow$ König John ist eine Person

Richards linkes Bein ist ein König $\Rightarrow$ Richards linkes Bein ist eine Person

John linkes Bein ist ein König $\Rightarrow$ John linkes Bein ist eine Person

Die Krone ist ein König $\Rightarrow$ Die Krone ist eine Person

P	Q	P $\Rightarrow$ Q
false	false	true
false	true	true
true	false	false
true	true	true

Die Implikation ist wahr, wenn ihre Prämisse falsch ist – unabhängig von der Wahrheit der Schlussfolgerung. Also sind die obigen 5 Sätze wahr, denn bis auf John ist keines der 5 Objekte ein König.

Vorsicht: Implikation und Konjunktion nicht verwechseln

Der Satz

$\forall$ x König(x) $\wedge$ Person(x)

ist äquivalent mit den Behauptungen:

Richard Löwenherz ist ein König $\wedge$ Richard Löwenherz ist eine Person

König John ist ein König $\wedge$ König John ist eine Person

Richards linkes Bein ist ein König $\wedge$ Richards linkes Bein ist eine Person.

usw.

Die Verwendung der Konjunktion als Hauptverknüpfung mit dem Allquantor führt zu einer überstarken Aussage.

Die Implikation $\rightarrow$ ist die natürliche Verknüpfung (connection) für die Verwendung in Kombination mit dem Allquantor $\forall$

Existenzquantor

Mit dem **Existenzquantor $\exists$** kann eine Aussage zu einem bestimmten Objekt im Universum gemacht werden, ohne es per Namen zu nennen.

Um beispielsweise zu sagen, dass König John eine Krone auf dem Kopf hat, schreiben wir:

$\exists x$ *Krone(x)* $\wedge$ *AufDemKopf(x, John)*

Sprechweisen für $\exists x$:

„Es gibt ein x, so dass gilt ...“ oder „Für ein x ...“

Intuitiv besagt der Satz: $\exists x$ P, P ist mindestens für ein Objekt wahr.

Genauer: $\exists x$ P ist in einem bestimmten Modell unter einer bestimmten Interpretation wahr, wenn P in mindestens einer erweiterten Interpretation wahr ist, die x einem Domänenelement zuweist.

Existenzquantor und Konjunktion

Die natürliche Verknüpfung des Existenzquantors ist die Konjunktion.

Die Verwendung der Implikation als Hauptverknüpfung mit dem Existenzquantor führt zu einer sehr schwachen Aussage.

Betrachte den folgenden Satz:

$$\exists x\ Krone(x) \Rightarrow AufDemKopf(x, John)$$

Wendet man auf diesen Satz die Semantik an, so besagt der Satz, dass mindestens eine der folgenden Behauptungen wahr sei:

Richard Löwenherz ist eine Krone ⇒ Richard Löwenherz befindet sich auf dem Kopf von John

König John ist eine Krone ⇒ König John befindet sich auf dem Kopf von John

Richards linkes Bein ist eine Krone ⇒ Richards linkes Bein befindet sich auf dem Kopf von John

usw.

P	Q	P ⇒ Q
false	false	**true**
false	true	**true**
true	false	false
true	true	**true**

Verschachtelte Quantoren

Komplexe Sätze werden unter Verwendung mehrerer Quantoren ausgedrückt.

Einfacher Fall: alle Quantoren vom selben Typ:

∀x ∀y Bruder(x,y) ⇒ Geschwister(x,y)

Gemischte Fälle:

„Jeder liebt jemanden" bedeutet, dass es für jede Person jemanden gibt, den diese Person liegt.

∀x ∃y Liebt(x,y)

„Es gibt jemanden, der von jedem geliebt wird":

∃y ∀x Liebt(x,y)

Es kommt also auf die Reihenfolge der Quantoren an.

Verbindungen zwischen Quantoren

Die beiden Quantoren sind durch die Negation eng miteinander verknüpft.

$\forall x \neg Mag(x,Honig)$ ist äquivalent mit $\neg \exists x\ Mag(x, Honig)$

Die de Morganschen Regeln können für quantifizierte und nicht quantifizierte Sätze formuliert werden.

$\forall x\ \neg P$	$\equiv$	$\neg \exists x\ P$	$\neg P \wedge \neg Q$	$\equiv$	$\neg(P \vee Q)$
$\neg \forall x\ P$	$\equiv$	$\exists x\ \neg P$	$\neg P \wedge Q$	$\equiv$	$\neg P \vee \neg Q$
$\forall x\ P$	$\equiv$	$\neg \exists x\ \neg P$	$P \wedge Q$	$\equiv$	$\neg(\neg P \vee \neg Q)$
$\exists x\ P$	$\equiv$	$\neg \forall x\ \neg P$	$P \vee Q$	$\equiv$	$\neg(\neg P \wedge \neg Q)$

Gleichheit

Wenn sich zwei Terme auf dasselbe Objekt beziehen, so wird das **Gleichheitssymbol** verwendet.

$$Vater(John) = Henry$$

Um auszudrücken, dass Richard mindestens zwei Brüder hat, schreiben wir:

$$\exists x, y\ Bruder(x, Richard) \wedge Bruder(y, Richard) \wedge \neg(x = Y)$$

Anwendungen der FOL

TELL/ASK-Schnittstelle für eine Wissensbasis erster Stufe.

Mit TELL werden Sätze zu einer Wissensbasis hinzugefügt. Solche Sätze werden als **Zusicherung** bezeichnet.

TELL*(KB, König(John))*.

TELL*(KB, ∀x König(x) ⇒ Person(x)*.

Mit ASK fragen wir die Wissensbasis:

ASK(*KB, König(John)*).

ASK(*KB, ∃x Person(x)*).

Fragen, die mit ASK gestellt werden, werden als **Abfragen** oder **Ziele** (Goals) bezeichnet.

Die Verwandtschaftsdomäne

Wir definieren binäre Prädikate wie Eltern, Geschwister, Bruder, Schwester, Kind, etc. Wir verwenden Funktionen wie Mutter und Vater, weil jeder Mensch genau eines davon hat.

Die Mutter ist der weibliche Elternteil:

$\forall m, c\ Mother(c) = m \Leftrightarrow Female(m) \wedge Parent(m, c)$

Der Ehemann ist der männliche Elternteil:

$\forall w, h\ Husband(h, w) \Leftrightarrow Male(h) \wedge Spouse(h, w)$

Männlich und weiblich sind disjunkte Kategorien:

$\forall x\ Male(x) \Leftrightarrow \neg Female(x)$

Eltern und Kind sind inverse Relationen:

$\forall p, c\ Parent(p,c) \Leftrightarrow Child(c,p)$

Großvater-Relation:

$\forall g, c\ Grandparent(p,c) \Leftrightarrow \exists p\ Parent(g,p) \wedge Parent(p,c)$

Die Verwandtschaftsdomäne ...

Ein Geschwister ist ein weiteres Kind der Eltern:

$$\forall x, y\ Sibling(x,y) \Leftrightarrow x \neq y \wedge \exists p\ Parent(p,x) \wedge Parent(p,y)$$

usw.

Jeder dieser Sätze kann als **Axiom** der Verwandschaftsdomäne betrachtet werden. Die Verwandschaftsaxiome sind auch eine Menge von grundlegenden Definitionen. Sie definieren die Funktion *Mother* und die Prädikate *Husband*, *Male*, *Parent*, *Grandparent* und *Sibling*. Mit diesen können weitere Prädikate definiert werden.

Theoreme sind durch die Axiome folgerbar, z.B.

$$\forall x, y\ Sibling(x,y) \Leftrightarrow Sibling(y,x)$$

Nicht jedes Prädikat muss vollständig definiert werden. Es reichen partielle Spezifikationen von Eigenschaften. Beispiel *Person(x)*.

Zahlen, Mengen und Listen

Die Peano-Axiome definieren **natürliche Zahlen** und die Addition. Natürliche Zahlen sind rekursiv definiert:

$$NatNum(0)$$

$$\forall n\ NatNum(n) \Rightarrow NatNum(S(n))$$

Axiome, die die Nachfolgerfunktion *S(n)* beschränken:

$$\forall n\ \ 0 \neq S(n)$$

$$\forall m, n\ \ m \neq n\ \Rightarrow S(m) \neq S(n)$$

Definition der Addition mit Hilfe der Nachfolgerfunktion:

$$\forall m\ NatNum(m) \Rightarrow +(m,\ 0) = m$$

$$\forall m, n\ \ NatNum(m) \wedge NatNum(n) \Rightarrow +(S(m),\ n) = S(+(m,\ n))$$

S(n) kann auch als $n+1$ geschrieben werden (Infixnotation):

$$\forall m, n\ \ NatNum(m) \wedge NatNum(n) \Rightarrow (m+1)+n = (m+n)+1$$

Zahlen, Mengen und Listen

Mögliche Axiome zur Definition von Mengen:

1. Die einzigen Mengen sind die leere Menge und die Mengen, die entstehen, indem man einer Menge etwas hinzufügt:

$$\forall s\ Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2\ \ Set(s_2) \wedge s = \{x \mid s_2\})$$

2. Der leeren Menge wurden keine Elemente hinzugefügt, es gibt also keine Möglichkeit, *EmptySet* in eine kleinere Menge zu zerlegen:

$$\neg\exists\ x, s\ \ \{x \mid s\} = \{\ \}$$

3. Das Hinzufügen eines Elemente, das sich bereits in der Menge befindet, hat keine Wirkung:

$$\forall x, s\ \ x \in s \Leftrightarrow s = \{x \mid s\}$$

...

Zahlen, Mengen und Listen

Listen sind mit Mengen vergleichbar. Der Unterschied ist, dass Listen sortiert sind und in einer Liste dasselbe Element mehrfach erscheinen kann.

Die Wumpuswelt ...

Mit Hilfe der FOL können wir die Wissensbasis der Wumpus-Welt sehr viel knapper und in natürlicherer Weise definieren als in der Aussagenlogik.

Ein typischer Wahrnehmungssatz enthält auch die Zeit, wan die Wahrnehmung aufgetreten ist:

Percept([*Stench*, *Breeze*, *Glitter*, *None*, *None*], 5)

Aktionen können durch logische Terme repräsentiert werden:

Turn(Right), Turn(Left), Forward, *Shoot*, *Grab*, *Release*, *Climb*

Um zu erkennen, welche Aktion die beste ist, erstellt das Agentenprogramm eine Abfrage wie etwa:

∃ *a* *BestAction(a*, 5)

ASK soll diese Abfrage auflösen und eine Bindungsliste wie etwas {*a* | *Grab*} zurückgeben.

Die Wumpuswelt ...

Die reinen Wahrnehmungsdaten implizieren bestimmte Fakten über den aktuellen Zustand, zum Beispiel:

∀ *t, s, g, m, c* *Percept*([*s*, *Breeze*, *g, m, c*], *t*) ⇒ *Brezze(t)*

∀ *t, s, g, m, c* *Percept*([*s*, *b*, *Glitter, m, c*], *t*) ⇒ *Glitter(t)*

...

Durch quantifizierte Implikationssätze kann auch einfaches „Reflexverhalten“ implementiert werden. Zum Beispiel:

∀ *t Glitter(t)* ⇒ *BestAction(Grab, t)*

Die Nachbarschaft von zwei Quadraten kann definiert werden als:

∀ x,y,a,b Adjacent([x,y], [a,b]) ⇔ [a,b] ∈ {[x+1,y], [x-1,y], [x,y+1], [x, y-1]}

Die Position des Agenten ändert sich über der Zeit; *At(Agent, s, t)*, d.h. der Agent befindet sich zur Zeit t aus Quadrat t. Wir schreiben:

∀ s, t At(Agent, s, t) ∧ Breeze(t) ⇒ Breezy(s)

Diagnoseregeln in der Wumpuswelt

Diagnoseregelen führen von beobachteten Effekten zu verborgenen Dingen. Um Falltüren zu finden, besagen die Diagnoseregeln, dass, wenn ein Quadrat windig ist, ein benachbartes Quadrat eine Falltür enthalten muss:

$$\forall s \; Breezy(s) \Rightarrow \exists r \; Adjacent(r,s) \wedge Pit(r)$$

Und die Negation gilt auch:

$$\forall s \; \neg Breezy(s) \Rightarrow \neg\exists r \; Adjacent(r,s) \wedge Pit(r)$$

Kombiniert man diese beiden Regeln, so folgt:

$$\forall s \; Breezy(s) \Leftrightarrow \exists r \; Adjacent(r,s) \wedge Pit(r)$$

Kausale Regeln in der Wumpuswelt

Kausale Regeln reflektieren die angenommene Richtung in der Welt: Einige verborgene Eigenschaften der Welt bewirken, dass bestimmte Wahrnehmungen erzeugt werden. Beispielsweise bewirkt eine Falltür, dass alle benachbarten Quadrate windig sind:

$$\forall r \; Pit(r) \; [\forall s \; Adjacent(r,s) \Rightarrow \mathrm{Breezy}(s)$$

Wenn alle zu einem bestimmten Quadrat benachbarten Quadrate windfrei sind, ist das Quadrat nicht windig:

$$\forall s \; [\forall r \; Adjacent(r,s) \Rightarrow \neg Pit(r)] \Rightarrow \neg Breezy(s)$$

Diese beiden Sätze sind äquivalent mit dem bikonditionalen Satz (siehe vorherige Folie):

$$\forall s \; Breezy(s) \Leftrightarrow \exists r \; Adjacent(r,s) \wedge Pit(r)$$

Modellbasierte Schlusssysteme

Systeme, die mit kausalen Regeln schließen, werden als **modellbasierte Schlusssysteme** bezeichnet, weil die kausalen Regeln ein Modell dessen bilden, wie die Umgebung arbeitet.

Wenn die Axiome die Arbeitsweise der Welt und die Erstellung der Wahrnehmungen korrekt und vollständig beschreiben, leitet jede vollständige Inferenzprozedur die stärkste mögliche Beschreibung des Weltzustands für die verfügbaren Wahrnehmungen ab.